

The Cleaning Robot Project

Aplicação do Filtro de Kalman na Auto-Localização de um Sistema Robótico Autónomo

Fernando Pedro Vieira Freitas Pinto

Dissertação submetida para a obtenção do grau de
Mestre em Engenharia Electrotécnica e de Computadores
pela Faculdade de Engenharia da Universidade do Porto

Trabalho realizado sob a orientação do
Professor Doutor Armando Jorge Miranda de Sousa

Presidente do Júri

Universidade do Porto
Faculdade de Engenharia
Departamento de Engenharia Electrotécnica e de Computadores

Abril 2008

Este projecto foi realizado com o apoio de:



Universidade do Porto

Faculdade de Engenharia

FEUP

DEEC

Departamento de Engenharia Electrotecnia e de Computadores da Faculdade de
Engenharia da Universidade do Porto



Institute for
Systems & Robotics
Porto

Institute of Systems and Robotics - Porto



Futebol Robótico da Faculdade de Engenharia da Universidade do Porto

Gostaria agradecer

A minha família que sempre me ajudou,
especialmente a minha mãe e irmã que são o centro da minha vida,
e meu pai, que do Céu me protege e está sempre ao meu lado.

Aos meus verdadeiros amigos que sempre me apoiaram.

Aos meus colegas de curso e em
especial ao meu companheiro de muitos e muitos trabalhos Gustavo Pimentel,
e ao Pedro Machado, José Barros por todo o seu apoio.

Ao meu orientador do projecto Armando Sousa
pela dedicação, confiança e horas perdidas.

A Todos Muito Obrigado

Fernando Pedro Vieira Freitas Pinto
Março 2008

"...

Eras sobre eras se somem

No Tempo que em eras vem

Ser descontente é ser homem

Que as forças cegas se domem

Pela visão que a alma tem!

..."

Fernando Pessoa,

excerto do poema O Quinto Império,

in Mensagem

Resumo

Os sistemas robóticos como um todo tem sido foco de atenção e de muito desenvolvimento desde o último século, passando do imaginário das pessoas para a realidade e para a coabitação no dia a dia, seja em casa, ou no trabalho.

No caso dos sistemas robóticos móveis, uma das áreas com maior desenvolvimento e que tem chamado mais a atenção da comunidade científica é a dos chamados robot de serviço, robot estes, desenvolvidos exclusivamente para nos auxiliar nas mais variadas funções do dia a dia, tirando-nos da realização de funções consideradas sujas, perigosas ou não produtivas.

Uma característica obrigatória neste tipo de robot é que sejam autónomos, necessitando para isso de saberem decidir e processar informação diversa, de modo a realizarem as suas tarefas de forma eficiente, precisa e em segurança. Uma outra necessidade obrigatória é a capacidade de mobilidade, ou seja, o robot tem que ser capaz de se deslocar livremente num ambiente conhecido, necessitando para isso de conhecer a sua posição no mundo.

A problemática do "Onde é que estou?" num sistema robótico é muito complexa, sendo para isso utilizados sistemas de fusão de informação, como o Filtro de Kalman, que através de informação sensorial distinta, consegue de uma forma matemática, realizar uma estimação do estado robot, ou seja, estimar a sua posição em termos de X , Y e θ .

Nesta dissertação é analisado da aplicação do Filtro de Kalman na auto-localização de um robot autónomo de limpeza a operar num ambiente dinâmico e semi-estruturado.

Abstract

The robotic systems as a whole has been the focus of attention and the most developed since the last century, through the imagination of people to the reality and the cohabitation on a daily basis, whether at home or at work.

In the case of mobile robotic systems, one of the areas with more development which has large attention from the scientific community because the so-called service robot. Robot that was developed exclusively to help in the most varied duties on a daily bases, taking us Humans from carrying out functions considered dirty, dangerous or not productive.

A mandatory feature in this type of robot is the autonomous, requiring it to decide, to learn and process different information, in order to carry out their tasks efficiently, accurately and reliably. Another need is the ability to mandatory mobility, the robot must be able to move freely in a known environment.

The issue of "Where am I?" in a robotic system is very complex one. It is used to respond to a need solution basead in Data Fusion Algorithms, such as the Kalman Filter. The Kalman Filter can process different sensory information, in a mathematic way , performing an sucessefull estimation of the state robot, giving and estimating the position in terms of X , Y and θ .

In this thesis is considered the application of the Kalman Filter in the self-location of an autonomous robot for cleaning, operating in a dynamic and semi-structured environment.

Índice

Resumo.....	v
Abstract.....	vi
Índice.....	vii
Índice de Figuras.....	x
Índice de Tabelas.....	xiii
Capítulo 1	
Introdução.....	1
1.1 Motivação e Contexto.....	1
1.2 Contribuições.....	3
1.3 Estrutura.....	3
Capítulo 2	
Controladores PID.....	4
2.1 Introdução.....	4
2.2 Controlo por Modelo Interno.....	5
2.2.1 Introdução.....	5
2.2.2 Método de Chien.....	6
Capítulo 3	
Filtro de Kalman.....	7
3.1 Introdução.....	7
3.2 Aplicações.....	9
3.3 Conceito.....	9
3.3.1 Estimador de Estado.....	10
3.3.2 Confiança no Estado.....	10
3.4 Pressupostos à aplicação do Filtro de Kalman.....	11

3.4.1 Sistema Dinâmico Linear.....	11
3.4.2 Características do Ruído.....	12
3.5 Implicações da Distribuição Gaussiana.....	14
3.5.1 Modelo do Sistema e das Medidas.....	14
3.5.2 Incerteza Gaussiana.....	16
3.5.3 Estimador Óptimo.....	17
3.6 Formulação Base do Filtro de Kalman.....	19
3.7 Filtro de Kalman Linear.....	21
3.7.1 Inicialização.....	21
3.7.2 Previsão.....	21
3.7.3 Actualização.....	21
3.8 Estimador de Variância Mínima.....	25
3.9 Extensões ao Filtro de Kalman.....	27
3.9.1 Introdução.....	27
3.9.2 Filtro de Kalman Discreto.....	27
3.9.3 Sistemas Dinâmicos Não Lineares.....	29
3.9.4 Filtro de Kalman Extendido.....	29
3.9.5 Filtro de Kalman Extendido Iterativo.....	34
3.10 Conclusões.....	36
Capítulo 4	
Trabalho Desenvolvido.....	38
4.1 O Robot de Limpeza.....	38
4.1.1 Apresentação Geral do Robot de Limpeza.....	39
4.2 Calibração dos PID's.....	48
4.2.1 Abordagem Implementada.....	48
4.2.2 Dados Experimentais.....	50
4.2.3 Conclusões.....	57
4.3 Calibração da Odometria.....	57
4.3.1 Abordagem Implementada.....	57
4.3.2 Dados Experimentais.....	59
4.3.3 Conclusões.....	62
4.4 Modelo do Erro dos Sonares.....	62
4.4.1 Abordagem Implementada	62
4.4.2 Conclusões.....	65
4.5 Arquitectura de Software.....	66
4.6 Auto-localização por Filtro de Kalman.....	71
4.6.1 Abordagem Implementada.....	71
4.6.2 Simulações.....	72
4.6.3 Dados Experimentais.....	90
4.7 Conclusões.....	103
Capítulo 5	
Conclusões.....	105
5.1 Conclusões Gerais.....	105
5.2 Trabalho Futuro.....	106

Referências Bibliográficas.....	108
Anexo - Dados Experimentais da Calibração da Odometria.....	cx
Anexo - Ficha Técnica.....	cxiii

Índice de Figuras

Figura 2.1: Modelo por Blocos de um Controlador PID.....	5
Figura 4.1: Imagens Actuais do Robot de Limpeza.....	39
Figura 4.2: Robot com Locomoção Diferencial.....	40
Figura 4.3: O Antes e o Depois do Robot de Limpeza.....	40
Figura 4.4: Arquitectura Tecnológica do Robot de Limpeza.....	42
Figura 4.5: Fotografia dos Drivers utilizados para o controlo e alimentação dos motores do Robot de Limpeza.....	44
Figura 4.6: Mapa do Segundo Piso do Edifício Poente do Departamento de Engenharia Electrotécnica da Universidade do Porto, com a disposição dos códigos barras pela área.....	45
Figura 4.7: Imagem do Programa de Medição de Distâncias a Códigos de Barras... 46	
Figura 4.8: Sonares SRF08.....	47
Figura 4.9: Difusão Teórica para o Eco do Sonar.....	47
Figura 4.10: Dispositivo de medição de distâncias por infra-vermelhos (Sharp GP2Y0A02YK0F).....	48
Figura 4.11: Esquema do Controlador PID implementado nos drivers de controle dos Motores.....	49
Figura 4.12: Referências de Velocidade Aplicadas na determinação dos Parâmetros do PID.....	50
Figura 4.13: Gráfico da Resposta em Malha Aberta do Motor da Direita, em termos de Velocidade.....	52
Figura 4.14: Resposta em Malha Aberta, Motor da Direita, em termos de Corrente.....	53
Figura 4.15: Resposta em Malha Fechada, Motor da Direita, em termos de Velocidade.....	54
Figura 4.16: Resposta em Malha Fechada, Motor da Direita, em termos de Corrente.....	54
Figura 4.17: Resposta em Malha Aberta, Motor da Esquerda, em termos de Velocidade.....	55
Figura 4.18: Resposta em Malha Aberta, Motor da Esquerda, em termos de	

Corrente.....	56
Figura 4.19: Resposta em Malha Fechada, Motor da Esquerda, em termos de Velocidade.....	57
Figura 4.20: Resposta em Malha Fechada, Motor da Esquerda, em termos de Corrente.....	57
Figura 4.21: Esquema utilizado para a Calibração da odometria e Obtenção do Perímetro das Rodas.....	59
Figura 4.22: Esquema utilizado para a Calibração da odometria e Obtenção da Distância entre Rodas.....	59
Figura 4.23: Modelo para a Determinação do erro de Translação e do Deslizamento.....	60
Figura 4.24: Representação Gráfica dos dados Experimentais para o Modelo de Translação e Deslizamento.....	61
Figura 4.25: Esquema utilizado para a obtenção do Modelo do Erro na Rotação	62
Figura 4.26: Representação Gráfica dos Dados Experimentais para o Modelo de Rotação.....	63
Figura 4.27: Grelha da disposição do Objecto Cilíndrico de Testes em relação ao Sonar.....	64
Figura 4.28: Grelha da disposição do objecto plano de testes em relação ao Sonar.....	64
Figura 4.29: Representação gráfica dos erros absolutos em relação à distância no teste com um cilindro de acrílico.....	64
Figura 4.30: Representação gráfica dos erros absolutos em relação à distância no teste com uma placa de contra-placado.....	65
Figura 4.31: Principais Eventos presentes no Programa de Controlo e Decisão do Robot de Limpeza.....	67
Figura 4.32: Esquema em Blocos das acções a tomar no Ciclo de Controlo.....	68
Figura 4.33: Mecanismos de Auto-Localização no Robot de Limpeza.....	69
Figura 4.34: Interface Gráfica da Aplicação Log Replay do Projecto do Robot de Limpeza.....	71
Figura 4.35: Interface de Time Line Control da Aplicação Log Replay.....	71
Figura 4.36: Acções a Realizar durante a Estimação via Filtro de Kalman Extendido.....	72
Figura 4.37: Entrada (força => aceleração) ao longo do tempo.....	75
Figura 4.38: Gaussianas representativas de Posição e Velocidade - Cenário 1...	75
Figura 4.39: Comparação entre a posição e velocidade estimados e “reais” - Cenário 1.....	76
Figura 4.40: Erros entre a estimativa e medida “real” para posição e velocidade - Cenário 1.....	76
Figura 4.41: Gaussianas representativas de Posição e Velocidade - Cenário 2...	77
Figura 4.42: Comparação entre a posição e velocidade estimados e “reais” - Cenário 2.....	78
Figura 4.43: Erros entre a estimativa e medida “real” para posição e velocidade - Cenário 2.....	78
Figura 4.44: Entrada do sistema.....	80
Figura 4.45: Gaussianas representativas da posição do robot - cenário 1.....	81

Figura 4.46: Covariâncias da Posição Global do Robot - Cenário 1.....	82
Figura 4.47: Covariâncias da Posição do Robot XX - Cenário 1.....	82
Figura 4.48: Covariâncias da Posição do Robot YY - Cenário 1.....	83
Figura 4.49: Covariâncias da Posição do Robot em Theta - Cenário 1.....	83
Figura 4.50: Representação simulada no mapa - cenário 1.....	84
Figura 4.51: Representação simulada no mapa - cenário 1.....	85
Figura 4.52: Representação simulada no mapa - cenário 1.....	85
Figura 4.53: Gaussianas representativas da posição no cenário 2.....	86
Figura 4.54: Covariâncias da Posição Global do Robot - Cenário 2.....	87
Figura 4.55: Covariâncias da Posição do Robot XX - Cenário 2.....	87
Figura 4.56: Covariâncias da Posição do Robot YY - Cenário 2.....	88
Figura 4.57: Covariâncias da Posição do Theta do Robot - Cenário 2.....	88
Figura 4.58: Representação simulada no mapa - Cenário 2.....	89
Figura 4.59: Representação simulada no mapa - cenário 2.....	89
Figura 4.60: Representação simulada no mapa - cenário 2.....	90
Figura 4.61: Representação do Robot em Movimentação pelo Mundo.....	91
Figura 4.62: Covariância da Estimativa da Posição do Robot em XX - Dados Práticos de Odometria	93
Figura 4.63: Covariância da Estimativa da Posição do Robot em YY - Run de Odometria.....	93
Figura 4.64: Covariância da Estimativa da Posição do Robot em YY - Dados Práticos de Odometria.....	94
Figura 4.65: Representação da Estimativa da Posição do Robot e Covariâncias Associadas - Dados Práticos de Odometria.....	94
Figura 4.66: Representação do Robot a avistar um Código de Barras.....	96
Figura 4.67: Covariância da Estimativa da Posição do Robot em XX - Dados Práticos com Localização por Códigos de Barras.....	98
Figura 4.68: Covariância da Estimativa da Posição do Robot em YY - Dados Práticos com Localização por Códigos de Barras.....	98
Figura 4.69: Representação da Estimativa da Posição do Robot e Covariâncias Associadas - Dados Práticos com Localização por Códigos de Barras.....	99
Figura 4.70: Representação do Robot a realizar medidas com os Sonares.....	100
Figura 4.71: Representação do Robot a obter medidas dos Sharps.....	102

Índice de Tabelas

Tabela 2.1: Formulas de Cálculo das Constante do PID referentes ao Método Chien.....	6
Tabela 3.1: Resumo das operações do Filtro de Kalman Linear.....	25
Tabela 3.2: Resumo das operações do Filtro de Kalman Discreto.....	28
Tabela 3.3: Resumo das operações do Filtro de Kalman Extendido.....	34
Tabela 4.1: Modelo de Primeira Ordem com Atraso, definido para a calibração dos parâmetros do PID pelo Método de Chien.....	50
Tabela 4.2: Resumo dos Principais Dados obtidos sobre a Dinâmica do Motor da Direita.....	52
Tabela 4.3: Resumo dos Dados obtidos para a Calibração do Controlador PID, segundo o Método de Chien, para o Motor da Direita.....	53
Tabela 4.4: Parâmetros do PID utilizados para o Motor da Direita.....	53
Tabela 4.5: Resumo dos Principais Dados obtidos sobre a Dinâmica do Motor da Esquerda.....	55
Tabela 4.6: Resumo dos Dados obtidos para a Calibração do Controlador PID, segundo o Método de Chien, para o Motor da Direita.....	56
Tabela 4.7: Parâmetros do PID utilizados para o Motor da Esquerda.....	56
Tabela 4.8: Modelo de odometria - Erro de Translação.....	61
Tabela 4.9: Modelo de odometria - Erro de Deslizamento.....	61
Tabela 4.10: Modelo de odometria - Erro de Rotação.....	62
Tabela 4.11: Modelo do Erro dos Sonares.....	65
Tabela 5.1: Dados Experimentais para a formulação do Modelo da Odometria relativos à Translação e Deslizamento.....	cxii
Tabela 5.2: Erros na Translação.....	cxii
Tabela 5.3: Erros de Deslizamento.....	cxii
Tabela 5.4: Dados Experimentais para a Formulação do Modelo da odometria Relativos à Rotação.....	cxiii

Capítulo 1

Introdução

1.1 Motivação e Contexto

Desde sempre que o homem procura a perfeição, uma busca constante de novas descobertas, por expandir os limites do conhecimento.

Pensadores de todas as fases da humanidade, desde Aristóteles, Leonardo Da Vinci, George Devol, escreveram e pensaram em sistemas automáticos e/ou autónomos que ajudassem na realização de tarefas difíceis ou perigosas para nós.

Foi com essa busca insaciável de conhecimento que apareceu a robótica, ou seja a ciência por trás da criação de dispositivos electromecânicos ou biomecânicos capazes de realizar trabalhos de uma forma autónoma, pré-programada ou mesmo com auxílio humano, ou seja, por trás da criação de robot.

A palavra robótica foi pela primeira vez utilizada numa pequena história de Issac Asimov "Runaround", publicada em Março de 1942, baseada na palavra robot, criada pelo autor de ficção científica Karel Capek.

Um dos ramos da robótica é a robótica móvel, e dentro desta, os robot de

serviço tem tipo uma evolução muito importante.

Estes normalmente desempenham acções de auxílio a trabalhos realizados por seres humanos.

Actualmente a actuação deste tipo de robot contempla diversas aplicações, como:

1. onde exista risco de vida significativo para humanos (no espaço, uso militar ou mesmo em ambientes hostis);
2. em funções economicamente desvantajosas no uso de trabalhadores humanos (indústria ou agricultura);
3. uso humanitário (cadeira de rodas, em hospitais ou em salvamentos);
4. uso educacional (são consideradas uma ferramenta poderosa para o ensino de engenharia).[1]

O trabalho apresentado nesta dissertação insere-se no projecto do Robot de Limpeza do DEEC, projecto este que foi iniciado antes do presente trabalho e terá futuro, existindo mesmo já novos elementos a trabalhar, que irão continuar onde esta dissertação parou, sempre com o intuito de melhorar a plataforma, procurando soluções cada vez mais inovadores, seja em termos de sensores, fusão de informação ou mesmo "Path Planning".

O principal objectivo desta dissertação corresponde a tornar o projecto do Robot de Limpeza, na realidade uma robot autónomo, ou seja, um robot capaz de tomar algumas decisões próprias, decisões essas necessárias para executar as suas tarefas com eficiência, precisão e segurança.

Uma especificação que se encontra sempre lado a lado com a característica de autonomia num sistema robótica é a sua mobilidade. Um robot com a capacidade de se deslocar livremente num ambiente conhecido, pode realizar de forma mais eficiente as acções para as quais foi desenvolvido.

Tendo isto em mente, uma das necessidades óbvias será a localização do robot no ambiente que o rodeia.

Para se conseguir isso, é necessário obter informação sensorial sobre o ambiente e depois utilizar essa informação, de modo racional, a fim de se conseguir uma estimativa para a posição do robot no mundo.

Em suma é necessário fundir a informação obtida.

<<**Fusão de Informação** é um quadro formal de trabalho onde se emprega meios e ferramentas para a junção de dados de origens diversas com o objectivo de obter informação de qualidade superior. A definição exacta de qualidade superior depende da aplicação em causa.>>[2]

1.2 Contribuições

Esta dissertação apresenta as seguintes contribuições de relevo:

- Apresentação abrangente do Filtro de Kalman como Sistema de Fusão de Informação;
- Definição de modos de calibração dos módulos principais num sistema robótico;
- Definição de uma Arquitectura de Controlo e Decisão robusta e com capacidade de análise de dados de modo "offline", replicando completamente todos os estados do sistema;
- Construção prática de um Sistema de Fusão de Informação, baseado no Filtro de Kalman Extendido, com apresentação e discussão de dados reais;
- Demonstração da viabilidade das soluções apresentadas para a localização robótica em ambientes estruturados.

1.3 Estrutura

A tese encontra-se dividida em cinco capítulos.

Inicialmente no Capítulo 1 é abordado o tema da tese e as suas contribuições.

No Capítulo 2, realiza-se uma breve apresentação teórica sobre controladores PID, especialmente na nos métodos de calibração baseados em Controlo por Modelo Interno.

No Capítulo 3, é apresentada a teoria por trás do Filtro de Kalman como método de fusão de informação. Explicam-se as suas bases, pressupostos e características de convergência. Apresenta-se ainda algumas das principais extensões especialmente para sistemas não lineares.

No Capítulo 4, é feito uma retrospectiva sobre todo o trabalho realizado, começando por uma apresentação do Robot de Limpeza, seguido da apresentação das principais calibrações sensoriais realizadas, explicação do sistema de controlo e decisão e todas as funcionalidades que este apresenta e proporciona e por fim a o Filtro de Kalman, tanto na sua vertente de simulação como implementação real no Robot de Limpeza.

Por último o Capítulo 5, onde são apresentadas conclusões e propostas de melhoramentos futuros.

Capítulo 2

Controladores PID

2.1 Introdução

O controlador proporcional-integral-derivativo (PID) é um mecanismo genérico de controlo de processos em malha fechada, muito utilizado em sistemas de controlo a nível industrial. É interessante, notar que mais de metade dos controladores indústrias em uso actualmente, empregam esquemas de controlo PID.

O controlor PID tenta corrigir o erro entre a medida do processo e a referência desejada, calculando e aplicando uma acção correctiva no processo. Envolve três parâmetros separados: o Proporcional, o Integral e o Derivativo. De uma forma simples e intuitiva posso explicar a contribuição de cada um dos termos do seguinte modo: o termo Proporcional determina a reacção ao erro actual; o termo Integral, a reacção, tendo em conta a soma dos mais recentes erros e, por fim, o Derivativo que determina uma reacção tendo em conta a forma como o erro tem variado. A soma pesada das três acções é utilizada para ajustar o processo via um elemento de controlo, seja ele a posição de uma válvula de controlo, a energia fornecida a um aquecedor, etc.

A utilidade dos controladores PID está directamente relacionada com a

sua aplicabilidade geral na maioria dos sistemas de controlo. Em particular, quando o modelo matemático da planta não é conhecido, e portanto, o método de projecto analítico não pode ser utilizado, os controladores PID mostram-se muito úteis. Na área dos sistemas de controlo de processos, sabe-se que os esquemas básicos de controladores PID provaram a sua utilidade conferindo controlos satisfatórios, embora em muitas situações não proporcionem um controlo óptimo.

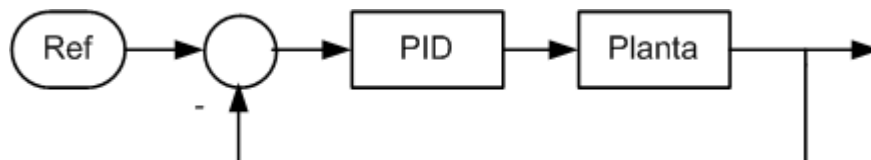


Figura 2.1: Modelo por Blocos de um Controlador PID

Nem sempre o controlador é usado na sua forma base, ou seja, como PID. Para que isto seja possível, basta colocar as componentes não desejadas a zero. Sendo assim, controladores do tipo PI, PD, P e I são igualmente possíveis.

Em termos de implementação real, o modo PI é muito comum, já que a acção derivativa é muito sensível ao ruído, o que leva que em muitas aplicações, o termo derivativo seja desprezado.

2.2 Controlo por Modelo Interno

2.2.1 Introdução

O controlo por modelo interno é apresentado como uma abordagem alternativa para a calibração clássica de PID, introduzida em 1982 por C. E. Garcia e M. Morari [3], mesmo não tendo sido pioneiros no estudo do método, realizaram e apresentaram avanços importantes.

O método tenta definir uma base de trabalho mais lógica e simples para o processo de calibração de PID, em que os procedimentos e os fundamentos teóricos básicos, possam ser facilmente assimilados.

Ao contrário dos modelos clássicos, o controlo por modelo interno, postula um modelo e define os objectivos para o controlo, procedendo-se depois de uma forma mais directa para a obtenção, tanto da estrutura apropriada para o controlador como dos parâmetros para o mesmo.

A eficiência do princípio do controlo por modelo interno (IMC) têm-no colocado numa posição cada vez mais atraente, no que contempla o controlo de processos industriais, tanto de processos estáveis como instáveis, também muito

fruto das constantes reclamações de controladores com erros de calibração, ou que não respondem a alguns objectivos desejados.

O modo de calibração IMC-PID tem ainda a vantagem de só utilizar um único parâmetro, conseguindo, com esse parâmetro, realizar um compromisso entre a performance do sistema em malha fechada e a robustez do sistema.

2.2.2 Método de Chien

Um método baseado no mesmo princípio, foi definido por I. L. Chien e P.S. Fruehauf em 1990 [4], onde os valores definidos para o controlador PID eram equivalentes, para sistemas com atraso, são apresentados na Tabela 2.1.

Tabela 2.1: Formulas de Cálculo das Constante do PID referentes ao Método Chien

Modelo do Processo	Parâmetros do Controlador		
	$k_c \cdot k$	τ_i	τ_D
$\frac{k_p \cdot e^{-Ls}}{1 + \tau s}$	$\frac{\tau}{\tau_{CL} + L}$	τ	
$\frac{k_p \cdot (\tau_3 s + 1) \cdot e^{-Ls}}{(\tau_1 s + 1) \cdot (\tau_2 s + 1)}$	$\frac{\tau_1 + \tau_2 - \tau_3}{\tau_{CL} + L}$	$\tau_1 + \tau_2 - \tau_3$	$\frac{\tau_1 \cdot \tau_2 - (\tau_1 + \tau_2 - \tau_3) \cdot \tau_3}{\tau_1 + \tau_2 - \tau_3}$
$\frac{k_p \cdot (-\tau_3 s + 1) \cdot e^{-Ls}}{(\tau_1 s + 1) \cdot (\tau_2 s + 1)}$	$\frac{\tau_1 + \tau_2 + \frac{\tau_3 \cdot L}{\tau_{CL} + \tau_3 + L}}{\tau_{CL} + \tau_3 + L}$	$\tau_1 + \tau_2 + \frac{\tau_3 \cdot L}{\tau_{CL} + \tau_3 + L}$	$\frac{\tau_3 \cdot L}{\tau_{CL} + \tau_3 + L} + \frac{\tau_1 \cdot \tau_2}{\tau_1 + \tau_2 + \frac{\tau_3 \cdot L}{\tau_{CL} + \tau_3 + L}}$
$\frac{k_p \cdot e^{-Ls}}{s}$	$\frac{2 \cdot \tau_{CL} + L}{(\tau_{CL} + L)^2}$	$2 \cdot \tau_{CL} + L$	

Capítulo 3

Filtro de Kalman

3.1 Introdução

Em 1960, R. E. Kalman publicou um artigo científico intitulado de “A New Approach to Linear Filtering and Prediction Problems”[5], com o intuito de ultrapassar as limitações do filtro de Wiener-Holf [6], na resolução de problemas de natureza estatística, que limitava a sua aplicabilidade prática. Este novo processo descrito R. E. Kalman ficou conhecido por Filtro de Kalman.

De um modo breve, o Filtro de Kalman pode ser descrito como um algoritmo recursivo de processamento de informação, proporcionando a estimação óptima do estado de um sistema dinâmico com ruído linear.

O filtro de Kalman tem as suas raízes na formulação, de sistemas dinâmicos, na forma de espaço de estados, o que leva a que quando se fala de estado, na realidade, estamos a falar de um vector de estado, normalmente representado pela letra X , composto por n variáveis que descrevem propriedades interessantes do sistema. Um exemplo para um vector de estado pode ser o apresentado na equação 3.1, que representa a localização de um robot, com as coordenadas de x e y e ainda a sua orientação θ .

$$X = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (3.1)$$

As variáveis de estado podem apresentar ruído ou mesmo não ser directamente observáveis, o que dificulta a estimação do estado. Para estimar o estado, o Filtro de Kalman tem acesso às medidas conseguidas pelo sistema, que devem estar linearmente dependentes do estado e apresentam algum ruído associado. Se as fontes de ruído apresentarem uma distribuição Gaussiana de média nula e com uma relação medida estado linear, então o Filtro de Kalman é um estimador de estado óptimo.[7]

O Filtro de Kalman é capaz de processar toda a informação disponível em termos de medidas para a realização da estimação do estado, sejam estas precisas ou imprecisas. Utiliza o conhecimento do sistema, modelo do sistema e da dinâmica dos sensores, relação entre as medidas e o estado, assim como da descrição probabilística do ruído associados, tanto ao modelo como às medidas e ainda, qualquer informação em relação ao valor inicial do estado.

O método apresenta o nome de “Filtro” devido a realizar uma estimação óptima a partir de medidas ou estados ruidosos, mas o método consegue ainda, alterar o estado do sistema de forma a reflectir, não só as novas medidas, como os ruídos existentes, realizando assim uma actualização dinâmica do sistema.

O Filtro de Kalman tem por base diversas disciplinas, tais como: mínimos quadrados; teoria das probabilidades; sistemas dinâmicos; sistemas estocásticos e álgebra, e foi base de algumas das mais recentes disciplinas, como a Teoria de Controlo Moderno e a Teoria de Decisão Estatística.

Alguns autores definem o Filtro de Kalman, apresentado em 1960 por R. E. Kalman, como uma das maiores descobertas na história das teorias de estimação estatística e, possivelmente, uma das maiores em todo o século vinte, o que leva a que a sua metodologia, definida em 1960, seja ainda hoje aplicada em diversas áreas do conhecimento. O Filtro de Kalman, fez com que nós fizéssemos coisas que não podiam ser realizadas sem recorrer a ele. Em termos de tecnologia moderna, o Filtro de Kalman já foi considerado, tão indispensável para o desenvolvimento humano e científico como os chips electrónicos.[7] [8]

3.2 Aplicações

O Filtro de Kalman tem sido utilizado, desde a sua definição, num conjunto diverso de aplicações, muito por causa das suas capacidade e potencialidade, tais como: a sua capacidade de estimar estados passados, presentes e mesmo futuros, conseguindo resultados mesmo quando a precisão modelo do sistema é desconhecida; a possibilidade de funcionar com uma representação minimalista do ambiente em que está inserido; a sua ilimitada exactidão na medida do estado e devido a ser considerado computacionalmente leve.[9]

Mesmo assim, as principais áreas de aplicação, investigação e desenvolvimento têm sido nos campos do controlo e predição de sistemas dinâmicos, sendo a sua aplicação mais frequente como estimador de estado.

Quando nos encontramos a controlar um sistema dinâmico, é importante compreender e conseguir saber tudo o que está a ocorrer no sistema. Quando estamos perante um sistema complexo, conseguir medidas de todas as variáveis que controlam o sistema é muitas vezes impossível. O Filtro de Kalman é capaz de providenciar informação sobre essas variáveis, que não conseguem ser directamente medidas, estimando o seu valor de forma indirecta e utilizando medidas ruidosas. [10]

Na área do controlo, o Filtro de Kalman tem sido utilizado no controlo de linhas de produção, em aviões, navios, naves espaciais e robots, especialmente na área de navegação autónoma ou assistida. [10] [11]

Na área da predição o Filtro de Kalman tem sido aplicado especialmente na predição de eventos futuros em sistemas dinâmicos, que são difíceis ou mesmo impossíveis de serem realizados ou controladas por pessoas. Alguns exemplos de sistemas onde é aplicado são: previsão de correntes de rios depois de uma cheia, trajectórias de corpos celestes ou preços de produtos. [7] [8] [10]

3.3 Conceito

O Filtro de Kalman é um estimador de estado que funciona recorrendo a duas etapas, uma de previsão e outra de actualização. Isto significa que o Filtro de Kalman terá que processar uma probabilidade ou melhor, uma medida da certeza do estado, primeiro sobre a previsão do estado, obtida através do modelo dinâmico do sistema e, mais tarde, depois de actualizar o estado, com as medidas obtidas do sistemas.

3.3.1 Estimador de Estado

O principal objectivo do Filtro de Kalman é de estimar o verdadeiro estado do sistema em causa. Em particular, os resultados do Filtro de Kalman são sempre uma estimativa do estado e uma medida da certeza na estimativa realizada.

O processo de estimação do Filtro de Kalman é complexo especialmente devido a possibilidade de alteração do estado ao longo do tempo, à necessidade de processamento de medidas ruidosas e ainda, à realização da estimação do estado que pode não ser directamente observável. Para além de todos estes factores, o Filtro de Kalman ainda tem que conseguir realizar de uma forma óptima a informação que processa, de modo a conseguir a estimação óptima do estado.

3.3.2 Confiança no Estado

De uma forma formal, o Filtro de Kalman estima a probabilidade condicionada de o sistema se encontrar no estado x_k , dadas as medidas z_1, \dots, z_k . Podemos chamar a probabilidade de o sistema se encontrar no estado x_k , dadas as medidas z_1, \dots, z_k como confiança no estado.

$$Bel^- = P(X_k | z_1, \dots, z_k) \quad (3.2)$$

Podemos dividir a definição de confiança em duas partes, baseando na regra de Bayes, teorema de probabilidade total e nas definições de Markov, como a confiança do estado à priori $Bel^-(x_k)$ e a confiança no estado à posteriori $Bel^+(x_k)$. Chegando as equações (3.3) e (3.4):

$$\begin{aligned} Bel^-(x_k) &= P(x_k | z_1, \dots, z_{k-1}) \\ &= \int_{\Xi} P(x_k \vee x_{k-1}) Bel^+(x_{k-1}) dx_{k-1} \end{aligned} \quad (3.3)$$

$$\begin{aligned} Bel^+(x_k) &= P(x_k | z_1, \dots, z_{k-1}) \\ &= \frac{P(z_k | x_k) \cdot Bel^-(x_k)}{P(z_k | z_1, \dots, z_{k-1})} \end{aligned} \quad (3.4)$$

A confiança no estado à priori, é a probabilidade condicionada do estado x_k dado todas as medidas obtidas z até ao instante actual k .

A confiança no estado à posteriori, é a probabilidade condicionada do estado x_k dado todas as medidas obtidas z até e incluindo o instante actual k .

De modo a se poder processar a confiança no estado, temos que descobrir o modelo do sistema $P(x_k | x_{k-1})$ mas também, a relação entre o estado e as medidas $P(z_k | x_k)$. [10]

3.4 Pressupostos à aplicação do Filtro de Kalman

3.4.1 Sistema Dinâmico Linear

O Filtro de Kalman assume que, tanto o estado como as medidas, são descritas por sistemas dinâmicos lineares. Ou seja, tem que ser possível descrever, tanto o modelo do sistema dinâmico, como a relação entre o estado e as medidas, ao longo do tempo, através de equações lineares.

O Filtro de Kalman pressupõe um modelo linear, já que é consideravelmente mais leve, em termos computacionais, para além da abordagem linear conseguir responder adequadamente à maioria dos problemas de modelização.

Quando estamos perante um problema não linear, teremos que utilizar outro tipo de abordagem, especialmente técnicas de linearização que transformem sistemas não lineares em sistema lineares.

Modelo da Dinâmica do Sistema

O modelo do sistema relaciona o estado verdadeiro e a sua evolução temporal. O Filtro de Kalman precisa do modelo, de modo a realizar a predição do estado, partindo do pressuposto que o estado do sistema evolui segundo a equação linear 3.5:

$$x_k = Ax_{k-1} + w_{k-1} \quad (3.5)$$

O estado do sistema $x_k \in \mathbb{R}^n$ referente ao instante k depende do estado anterior x_{k-1} e de ruído presente.

A matriz A de dimensão $n \times n$, relaciona o estado do sistema no instante $k-1$ com estado actual k . Esta matriz A é definida sem a contabilização da influência do ruído. Por outro lado, o vector $w_{k-1} \in \mathbb{R}^n$ modeliza o ruído no sistema, assim como modeliza efeitos não modelizados na matriz A , mas que na mesma influenciam a dinâmica do sistema.

Modelo das Medidas

O modelo das medidas é responsável por modelizar a relação entre as medidas obtidas e o estado do sistema.

No Filtro de Kalman, o modelo da medida, quando existem medidas válidas, é utilizado para realizar a actualização do estado actual em concordância com as medidas obtidas, ou seja, é responsável por realizar correcções no estado previsto pelo Filtro de Kalman, através do modelo da dinâmica do sistema.

O Filtro de Kalman pressupõe que as medidas devem ser modelizadas por

qualquer equação que relacione linearmente o estado do sistema com a medida, como o definido na equação.

$$z_k = Hx_k + v_k \quad (3.6)$$

A verdadeira medida $z_k \in \mathbb{R}^m$ no instante k depende linearmente do estado do sistema x_k . A matriz H de dimensão $m \times n$ relaciona o estado x_k com a medida z_k , como no caso do modelo da dinâmica do sistema, a matriz H é definida tomando que os sensores não apresentam ruído. No mundo real não existem sensores perfeitos, sendo então necessário acrescentar ruído à medida, o que é realizado pelo vector $v_k \in \mathbb{R}^m$, que modeliza esses erros.

Processos de Markov

É de salientar que na equação 3.5 o estado x_k no instante k não depende de todos os outros estados e medidas dadas; e também que na equação 3.6 que dado x_k e as medidas z_k não dependem de todos os outros estados e medidas obtidas.

Estas propriedades definem o sistema em causa como um Processo de Markov.

3.4.2 Características do Ruído

Como mencionado e modelizado nos pontos anteriores, tanto o sistema como as medidas, apresentam uma componente de ruído.

Tanto o ruído associado ao modelo w_k na equação (3.5), como o ruído associado as medidas v_k na equação (3.6), são assumidas pelo Filtro de Kalman, como variáveis aleatórias independentes, com componentes em todas as frequências, ruído branco, de media nula e respeitando uma distribuição Gaussiana.

Para além destas características, é assumido também que, a posição inicial do sistema x_0 , referente ao instante de tempo $k=0$ é também independente e apresenta uma distribuição gaussiana.

Independência

O pressuposto da independência nas componente de erro promovem que o processo computacional envolvido na estimação do estado é muito mais fácil. Em geral, é possível assumir que o ruído no sistema e nas medidas é independente, ou seja, que a medida do ruído associado a um, não influencia o outro.

Ruído Branco

O pressuposto de todo o ruído presente ser ruído branco, é responsável por uma enorme simplificação da matemática, por detrás do Filtro de Kalman.

O ruído branco, é ruído que apresenta componentes em todas as frequências do espectro e que são completamente não correlacionadas com ele próprio em qualquer instante de tempo, menos no instante actual.[8] No instante de tempo actual, o ruído existe, ou não existe, acontecendo isto sempre em todos os instante de tempo.

Este pressuposto implica que o erro não está é correlacionado ao longo do tempo, ou seja, o conhecimento da quantidade de ruído, num instante de tempo, não ajuda na previsão da quantidade de ruído para um de um outro qualquer instante de tempo.

Média Nula

É assumido que os erros apresentam média nula, ou seja, que os erros no sistema e nas medidas são aleatórios.

O ruído pode ser classificado como, ruído sistemático ou ruído não sistemático ou aleatório.

O ruído sistemático é definido como o ruído, que constantemente se encontra a degradar o estado do sistema, ou então da medida. Ou seja, é um tipo de ruído tendencioso, muito vezes causado por má calibração de parâmetros. Por exemplo, quando se determina de uma forma incorrecta a medida do diâmetro da roda de um robot, iremos estar perante um erro sistemático na estimação da posição.

Em contraponto, encontra-se o erro não sistemático ou aleatório. Este não está relacionado com erros na calibração de parâmetros, já que algumas vezes é positivo e outras negativo, mas sempre apresentado médias nulas.

Distribuição Gaussiana

O pressuposto da distribuição Gaussiana é responsável pelo tratamento da amplitude do ruído. Ela define a quantidade de ruído pode ser modelizado por uma curva que se assemelha com um campainha.

O pressuposto para o uso de distribuições gaussianas é justificado se assumirmos que, tanto os sistemas, como as medidas, apresentam ruído causado por diversas fontes de ruídos mais pequenas e como a soma de fontes individuais e independentes de ruído, seja qual for a sua distribuição, dá sempre uma distribuição gaussiana, tem justificado este pressuposto.

Se o ruído apresentar uma distribuição gaussiana promove que computacionalmente o filtro seja mais rastreável.

Um outro motivo que sugere a conveniência em assumir o ruído como modelizável por distribuição gaussiana, é de que normalmente, só é possível determinar a primeira e a segunda ordem da função estatística das fontes de ruído, ou seja, a média e o desvio padrão. A maioria dos dispositivos apresentam unicamente o valor nominal da medida, conseguindo-se com este valor, e de uma forma experimental, determinar a média e a variância do erro. As

distribuições gaussianas são completamente definidas e caracterizadas através dessas duas medidas, média e variância. Assim sendo, é possível avaliar completamente toda a informação do ruído.

Com a característica da média nula e distribuição gaussiana para o ruído, é possível descrever a distribuição. Se definirmos $Q_k = E[(w_k)(w_k)^T]$ como a covariância do ruído do modelo do sistema no instante k e $R_k = E[(v_k)(v_k)^T]$ como a covariância do ruído agora em relação às medidas e também no instante k , podemos exprimir o ruído do sistema w_k e das medidas v_k como:

$$w_k \sim N(0, Q_k) \quad (3.7)$$

$$v_k \sim N(0, R_k) \quad (3.8)$$

onde $N(\mu, \Sigma)$ representa uma função gaussiana de média μ e covariância Σ .

As diagonais principais das matrizes Q_k e R_k contêm o valor da variância nos estados e nas medidas, respectivamente, os elementos fora da diagonal principal são zero, já que é considerado que os valores do ruído são independentes.

3.5 Implicações da Distribuição Gaussiana

3.5.1 Modelo do Sistema e das Medidas

O pressuposto, que as variáveis aleatórias envolvidas serem distribuições gaussianas, trazem algumas propriedades interessantes, já que são completamente caracterizadas por dois parâmetros, a média e covariância.

Este pressuposto facilita imenso o processamento da incerteza do estado à priori e à posteriori.

Modelo Gaussiano para o Sistema

Pretende-se determinar como o modelo do sistema representado na equação 3.5 se distribui, supondo que apresenta uma distribuição Gaussiana tal como $P(x_k|x_{k-1})$ necessária para processar a incerteza à priori da equação 3.3.

Como mencionado antes, a evolução do estado do sistema é modelizado por uma equação linear do tipo.

$$x_k = Ax_{k-1} + w_{k-1} \quad (3.9)$$

Na qual o ruído associado ao sistema w_{k-1} apresenta média nula e covariância Q_k como na equação 3.7.

Analisando o termo Ax_{k-1} da equação 3.9, como apresentado uma distribuição gaussiana com média Ax_{k-1} e covariância nula, então:

$$Ax_{k-1} \approx N(Ax_{k-1}, 0) \quad (3.10)$$

Sendo assim não existe nenhuma incerteza no termo x_{k-1} , tomando agora que a distribuição do modelo do sistema $P(x_k|x_{k-1})$ e somando estas duas gaussianas ficamos com uma nova gaussiana, que representa a probabilidade do modelo do sistema, sendo este representado por:

$$P(x_k|x_{k-1}) = N(Ax_{k-1}, Q_k) \quad (3.11)$$

Modelo Gaussiano para as Medidas

Do mesmo modo, é necessário determinar a característica probabilística do modelo das medidas, presente na equação 3.6, processando a incerteza a posteriori segundo a equação 3.4 e sabendo a distribuição de $P(z_k|x_k)$.

A medida é modelizada por:

$$z_k = Hx_k + v_k \quad (3.12)$$

onde o ruído associado à medida v_k apresenta distribuição gaussiana de média nula e covariância R_k como na equação 3.8.

Analisando o termo Hx_k como uma distribuição gaussiana com média Hx_k e incerteza nula, então:

$$Hx_k \approx N(Hx_k, 0) \quad (3.13)$$

Não existe incerteza no modelo já que este é condicionado por x_k .

Dado as duas distribuições gaussianas, podemos concluir que a probabilidade condicionada da observação z_k dado x_k é também uma distribuição gaussiana.

$$P(z_k|x_k) = N(Hx_k, R_k) \quad (3.14)$$

3.5.2 Incerteza Gaussiana

É bom recordar que a incerteza à priori é a incerteza no estado x_k dada toda a informação até ao instante k e que a incerteza a posteriori é a incerteza utilizando toda a informação incluindo o instante k .

Com o pressuposto Gaussiano, podemos expressar a incerteza de uma forma mais precisa.

Incerteza Gaussiana à Priori

Dado que as gaussianas são completamente caracterizadas pela sua média e covariância, a incerteza à priori é:

$$Bel^-(x_k) = N(\hat{x}_k^-, P_k^-) \quad (3.15)$$

onde \hat{x}_k^- corresponde a média e P_k^- à covariância, sendo ambos em relação a estimativa do erro à priori.

A estimativa à priori e a covariância correspondem, respectivamente, ao primeiro e ao segundo momento estatístico

$$\hat{x}_k^- = E[x_k | z_1, \dots, z_k] \quad (3.16)$$

$$P_k^- = E[(e_k^-)(e_k^-)^T | z_1, \dots, z_{k-1}] \quad (3.17)$$

onde e_k^- corresponde a estimativa à priori do erro para o estado do sistema.

$$e_k^- = x_k - \hat{x}_k^- \quad (3.18)$$

Incerteza Gaussiana à Posteriori

De modo similar, a incerteza à posteriori, apresenta uma distribuição segundo:

$$Bel^+(x_k) = N(\hat{x}_k^+, P_k^+) \quad (3.19)$$

onde \hat{x}_k^+ corresponde à média e P_k^+ à covariância do erro da estimativa à posteriori. Sendo assim, a estimativa do estado e o erro da covariância à posteriori são definidos por:

$$\hat{x}_k^+ = E[x_k | z_1, \dots, z_k] \quad (3.20)$$

$$P_k^+ = E[(e_k^+)(e_k^+)^T | z_1, \dots, z_k] \quad (3.21)$$

onde e_k^+ é a estimativa do erro à posteriori do estado estimado.

$$e_k^+ = x_k - \hat{x}_k^+ \quad (3.22)$$

Estimativa do estado e erro na covariância

As estimativas para o estado \hat{x}_k^- e \hat{x}_k^+ são vectores que dada a informação presente sobre o sistemas, apresentam-se como as estimativas mais correctas para o estado real x_k do sistema.

As matrizes da covariância do erro P_k^- e P_k^+ contêm a covariância entre os diversos elementos do estado estimado, em particular, os elementos da diagonal principal destas matrizes contêm as variâncias dos elementos do estado, mas de uma forma individual. As variâncias presentes indicam a precisão da estimação do estado, realizada pelo Filtro de Kalman. Quanto menor forem os seus valores, mais precisa é a estimativa.

As estimativa do estado apresentadas nas expressões 3.16 e 3.20, para além das covariâncias apresentadas em 3.17 e 3.21, dependem das medidas aleatórias z_1, \dots, z_k que são variâncias aleatórias elas próprias.

Utilizando a definição do erro à posteriori, proveniente da equação 3.22 e a estimativa para o estado à posteriori expresso na equação 3.20, é possível processar a correlação.

$$\begin{aligned} E[e_k^+ z^k] &= E[x_k^k] - E[\hat{x}_k^+ z^k] \\ &= \int p(x_k, z^k) x_k z^k dx_k dz^k - \int \left(\int p(x_k | z^k) x_k dx_k \right) p(z^k) z^k dz^k \\ &= \int p(x_k, z^k) x_k z^k dx_k dz^k - \int p(x_k | z^k) p(z^k) x_k z^k dx_k dz^k \\ &= 0 \end{aligned} \quad (3.23)$$

A expressão é igual a zero porque:

$$p(x_k, z^k) = p(x_k | z^k) p(z^k) \quad (3.24)$$

Então, a correlação entre o erro e a medida desaparece, o que indica que estes dois factores são independentes, quando apresentam uma distribuição Gaussiana.

3.5.3 Estimador Óptimo

Antes de iniciar com a formulação teórica do Filtro de Kalman, achei por bem fazer uma rápida passagem pelos conceitos de estimação óptima, afim de apresentar de uma forma mais fundamentada, as características de estimador óptimo presentes no Filtro de Kalman.

A apresentação irá ser realizada num contexto de variáveis aleatórias escalares, já que, a generalização da teoria para vectores de variáveis aleatórias, é realizada de forma directa.

Suponhamos que nos é dada a observação:

$$y_k = x_k + v_k \quad (3.25)$$

onde x_k é um sinal desconhecido e v_k corresponde a adição de uma componente de ruído. Se \hat{x}_k for a estimativa à posteriori do sinal x_k , dadas as observações y_1, y_2, \dots, y_k . De um modo geral, a estimativa \hat{x}_k é diferente do sinal desconhecido x_k . Para realizar a estimativa de uma forma óptima, é necessário uma função de custo ou de degradação para estimativas incorrectas. A função de custo deverá satisfazer os seguintes requisitos:

- A função de custo deve ser não negativa;
- A função de custo é uma função não decrescente, para a estimação do erro x_k definida por:

$$\tilde{x}_k = x_k - \hat{x}_k \quad (3.26)$$

Estes dois requisitos, são satisfeitos pela determinação do erro, pelo método dos mínimos quadrados, definido por:

$$J_k = E[(x_k - \hat{x}_k)^2] \quad (3.27)$$

$$J_k = E[(\tilde{x}_k)^2] \quad (3.28)$$

onde E é o operador expectativa. A dependência da função de custo J_k no instante k enfatiza a natureza dinâmica do processo recursivo de estimação.

De modo a conseguir um valor óptimo para a estimativa \hat{x}_k , podemos invocar dois teoremas da teoria dos processos estocásticos.

Teorema 1: Estimador de Média Condicionada

Se os processos estocásticos $\{x_k\}$ e $\{y_k\}$ forem ambos Gaussianos, a estimativa óptima \hat{x}_k que minimiza o erro quadrático J_k é o estimador de Média Condicionada.

$$\hat{x}_k = E[x_k | y_1, y_2, \dots, y_k] \quad (3.29)$$

Teorema 2: Princípio da ortogonalidade

Se os processos estocásticos $\{x_k\}$ e $\{y_k\}$ apresentarem média nula, ou seja:

$$E[x_k] = E[y_k] = 0, \text{ para todo o } k \quad (3.30)$$

Então:

1. os processos estocásticos $\{x_k\}$ e $\{y_k\}$ são ambos Gaussianos; ou
2. se a estimativa óptima \hat{x}_k for restringida a uma função linear de observações e a função custo for o erro quadrático;
3. então a estimativa \hat{x}_k dada pelas observações y_1, y_2, \dots, y_k é a projecção ortogonal de x_k no espaço definido pelas observações.

3.6 Formulação Base do Filtro de Kalman

O Filtro de Kalman tem as suas raízes, como foi dito anteriormente, na formulação em espaço de estados de sistema de dinâmica linear, providenciando uma solução óptima recursiva para um problema de filtragem linear.

Consideremos um sistema genérico com dinâmica linear cujas saídas são combinações linear do estado:

$$\begin{aligned}\frac{d x(t)}{dt} &= A x(t) + B u(t) \\ y(t) &= C x(t)\end{aligned}\tag{3.31}$$

onde

$$x = [x_1 \ x_2 \cdots x_n]^T$$

$$u = [u_1 \ u_2 \cdots u_p]^T$$

$$y = [y_1 \ y_2 \cdots y_q]^T$$

$$\dim(A) = n \times n \ , \ m \text{ é a dimensão do sistema}$$

$$\dim(B) = p \times n \ , \ p \text{ é o número de entradas}$$

$$\dim(C) = q \times n \ , \ q \text{ é o número de saídas}$$

É possível converter o modelo em tempo contínuo apresentado acima, para um equivalente em tempo discreto:

$$\begin{aligned}x(k) &= \Phi x(k-1) + G u(k-1) \\ y(k) &= H x(k)\end{aligned}\tag{3.32}$$

em que o k é um número natural e representa o k -ésimo instante de amostragem.

Temos ainda que $\Delta t_k = t_k - t_{k-1}$ sendo possível demonstrar que:

$$\Phi(k) = e^{A \Delta t_k}\tag{3.33}$$

$$G = \int_{t_{k-1}}^{t_k} e^{A \Delta t_k} B dt\tag{3.34}$$

$$H = C\tag{3.35}$$

Na versão estocástica, é necessário adicionar incerteza na dinâmica do sistema $w(k)$ assim como, nas medidas $v(k)$, sendo o modelo agora representado por:

$$\begin{aligned}x(k) &= \Phi x(k-1) + Gu(k-1) + w(k-1) \\y(k) &= Hx(k) + v(k-1)\end{aligned}\tag{3.36}$$

De modo a garantir uma solução óptima e a convergência do método, $w(k)$ e $v(k)$ tem que ser processos Gaussianos com média nula além de ser necessário que sejam matrizes diagonais:

$$R_w(k, i) = \begin{cases} Q(k) & \text{se } i=k \\ 0 & \text{se } i \neq k \end{cases}\tag{3.37}$$

$$R_v(k, i) = \begin{cases} R(k) & \text{se } i=k \\ 0 & \text{se } i \neq k \end{cases}\tag{3.38}$$

$$E(w(k)v(i)^T) = 0, \forall k, i\tag{3.39}$$

$$\begin{aligned}P(w) &\sim N(0, Q) \\P(v) &\sim N(0, R)\end{aligned}\tag{3.40}$$

A matriz Q presente na expressão 3.37, representa a confiança na estimativa para o estado conseguida através do modelo. Por outro lado, a matriz R presente na expressão 3.38 representa a confiança do método nas medidas.

Em conjugação com as matrizes Q e R e das estimativas realizadas pelo método, é definida a matriz P , que representa a confiança global na estimativa. Os seus elementos da diagonal principal representam a covariância de cada um dos estados, enquanto os elementos fora da diagonal principal, expressão a covariância cruzada entre os estados. Esta matriz representa também uma noção de convergência do método, pois ao longo do tempo os valores devem ir diminuindo, já que cada vez devemos ter mais confiança nas estimativas realizadas.

3.7 Filtro de Kalman Linear

O Filtro de Kalman Linear foi o primeiro algoritmo do Filtro de Kalman a ser concebido e aplicado, saindo directamente da definição.

Consiste numa operação de inicialização, seguido de operações de previsão, de actualização executadas alternadamente.

3.7.1 Inicialização

O Filtro de Kalman Linear é inicializado com a estimativa inicial para o estado do sistema, x_0 , assim como da confiança que se tem nessa estimativa, através da matriz das covariâncias, P_0 .

3.7.2 Previsão

Nesta operação, o Filtro de Kalman realiza a estimação do estado do sistema no instante em causa, através da estimativa anterior para o estado do sistema e o modelo dinâmico do sistema.

$$x(k^-) = Ax(k-1) \quad (3.41)$$

A melhor estimativa, que é possível ser realizada pelo Filtro de Kalman após ter avançado um instante de tempo, é a propagação do estado anterior sobre o modelo dinâmico do sistema.

No Filtro de Kalman sabe-se sempre, que a evolução do sistema e mais propriamente, o seu modelo está sujeito a ruído, sendo assim, o valor da incerteza também tem que ser actualizada, baseando-se no erro do modelo e na incerteza do estado anterior.

$$P(k^-) = A P(k-1) A^T + Q(k-1) \quad (3.42)$$

O primeiro termo da expressão 3.42, $A P(k-1) A^T$ realiza a propagação da incerteza do instante anterior $P(k-1)$ para o instante actual, somando-lhe depois o ruído associado ao modelo do sistema $Q(k-1)$, ficando-se com um novo e actualizado valor para a matriz das covariâncias $P(k^-)$.

3.7.3 Actualização

Na actualização, o estado estimado na fase de previsão é actualizado com a informação das medidas. Sendo esta fase unicamente realizada, se novas medidas tiverem sido obtidas e validadas.

Para além de se actualizar o estado do sistema, é também actualizada a matriz das covariância.

É também nesta fase, que é determinado o chamado ganho de Kalman.

Este factor determina até que ponto se deve considerar as medidas obtidas, em relação à estimativa realizada na previsão.

Iniciemos pelo Ganho de Kalman, que é determinado através da matriz de covariâncias e da matriz das variâncias das medidas obtidas. Ou seja, relacionando, em termos de peso, a estimativa realizada pela previsão e as medidas obtidas pelo sistema.

$$K(k) = P(k^-) H^T (H P(k^-) H^T + R(k))^{-1} \quad (3.43)$$

De modo a conseguir-se comparar a incerteza definida pela previsão, com a incerteza nas medidas, é necessário converter a incerteza definida no espaço das medidas para o espaço do estado, utilizando-se para isso a matriz H^T .

Quando a covariância do erro das medidas R é aproximadamente zero, o ganho de Kalman coloca um peso elevado nas medidas obtidas. Ou seja, é considerado que as medidas apresentam mais informação para a estimação do estado, do que propriamente a informação da previsão, realizada através do modelo dinâmico e do estado anterior.

$$\lim_{R(k) \rightarrow 0} K(k) = H^{-1} \quad (3.44)$$

Neste caso, o Filtro de Kalman acredita menos no modelo do sistema, do que nas medidas.

Contrariamente, quando o erro determinado para o modelo $P(k^-)$ se aproxima de zero, o ganho de Kalman irá dar um maior peso à estimativa realizada na previsão do que às medidas conseguidas.

$$\lim_{P(k^-) \rightarrow 0} K(k) = 0 \quad (3.45)$$

Fazendo, com que o Filtro de Kalman se acredite mais na previsão, do que propriamente na actualização.

De seguida, é necessário então determinar a actualização das medidas em relação ao estado previsto, isso é conseguido através da expressão 3.46.

$$x(k) = x(k^-) + K(k)(y(k) - Hx(k^-)) \quad (3.46)$$

O termo $Hx(k^-)$ na expressão 3.46 corresponde à previsão da medida, dado a estimativa do estado $x(k^-)$ e a matriz do modelo das medidas H , o Filtro de Kalman consegue realizar a estimativa da medida que vai receber.

$$\hat{z}(k^-) = Hx(k^-) + v(k) \quad (3.47)$$

Para este cálculo, o ruído $v(k)$ é normalmente considerando zero, ficando com uma previsão para a medida que corresponde a uma variável aleatória com distribuição Gaussiana. Esta conclusão, é retirada devido à dependência linear do estado $x(k^-)$ e devido à medida do ruído, já que ambas são variáveis aleatórias Gaussianas. Sendo assim, podemos facilmente derivar a

previsão da medida:

$$\hat{z}(k) \sim N_z(Hx(k^-), H P(k^-) H^T + R(k)) \quad (3.48)$$

Agora é necessário analisar, o termo da medida de inovação, ou seja, a diferença entre a medida realmente obtida e a medida prevista.

$$\tilde{z}(k) = y(k) - \hat{z}(k) \quad (3.49)$$

Se a medida inovação for zero, então a previsão da medida reflecte exactamente a medida obtida. Isto implica que a estimativa do estado, com que a previsão da medida foi realizada, é muito próxima do estado real onde foram retiradas as medidas. Seja com for, quando a inovação não é nula, é necessário actualizar a estimativa do estado, o peso na actualização é definida pelo valor do ganho de Kalman.

A inovação depende da medida obtida $y(k)$ e da previsão da medida $\hat{z}(k)$. A incerteza do valor residual, não está dependente da medida obtida $y(k)$, que é considerada sem erro, mas sim na previsão da medida $\hat{z}(k)$ com distribuição gaussiana.

$$\begin{aligned} \tilde{z}_k &\sim N_z(\mu_{\tilde{z}_k, k}, \Sigma_{\tilde{z}_k, k}) \\ \mu_{\tilde{z}_k, k} &= y(k) - \hat{z}_k \end{aligned} \quad (3.50)$$

$$\Sigma_{\tilde{z}, k} = H P(k^-) H^T + R(k) \quad (3.51)$$

É de notar, que por vezes a própria medida real pode estar afectada de uma incerteza, devido, por exemplo, ao processo de extracção da medida que pode não se perfeito. Se isto ocorrer, então o valor residual calculado, tem que contabilizar também este facto.

Com a distribuição da inovação, é possível ter-se uma ideia da forma como este valor varia ao longo do tempo, dando também uma ideia dos erros associados à estimação da medida. Esta informação, pode ser utilizada para detectar inovações, que serão muito improváveis, dando a possibilidade de detectar ocasionalmente níveis de ruído exagerados nas medidas. Quando a informação da distribuição da inovação é utilizada para rejeitar medidas, dizemos que aplicamos uma porta de validação.[7]

Como descrito acima, quando a inovação é grande, indica que existe uma grande diferença entre a estimação para o estado real e para o estado do sistema.

Um modo simples de combinar a estimativa do estado e a inovação seria somar ambas as contribuições. No entanto, se a medida ou a estimativa da medida for muito ruidosa relativamente à estimativa da posição, esta solução de somar as duas contribuições não será a melhor, sendo neste caso, melhor ignorar a inovação.

No entanto, se as nossas medidas forem boas, ou mesmo menos ruidosas

do que as da estimativa do estado, então é considerável contabilizar com a inovação.

Na fase de actualização, para além da determinação do ganho de Kalman e da actualização do estado do sistema através da inovação apresentada pelas medidas, é necessário também, determinar a nova incerteza para o sistema, utilizando-se para isso a equação 3.52.

$$P(k) = (I - K(k)H)P(k^-) \quad (3.52)$$

Como podemos constatar da equação 3.52, o Filtro de Kalman utiliza o ganho de Kalman para determinar o novo valor para a matriz das incertezas.

Se voltarmos a analisar os casos de $R \rightarrow 0$ e $P(k^-) \rightarrow 0$ na determinação do valor para o Ganho de Kalman, equações 3.44 e 3.45 respectivamente, podemos concluir o seguinte:

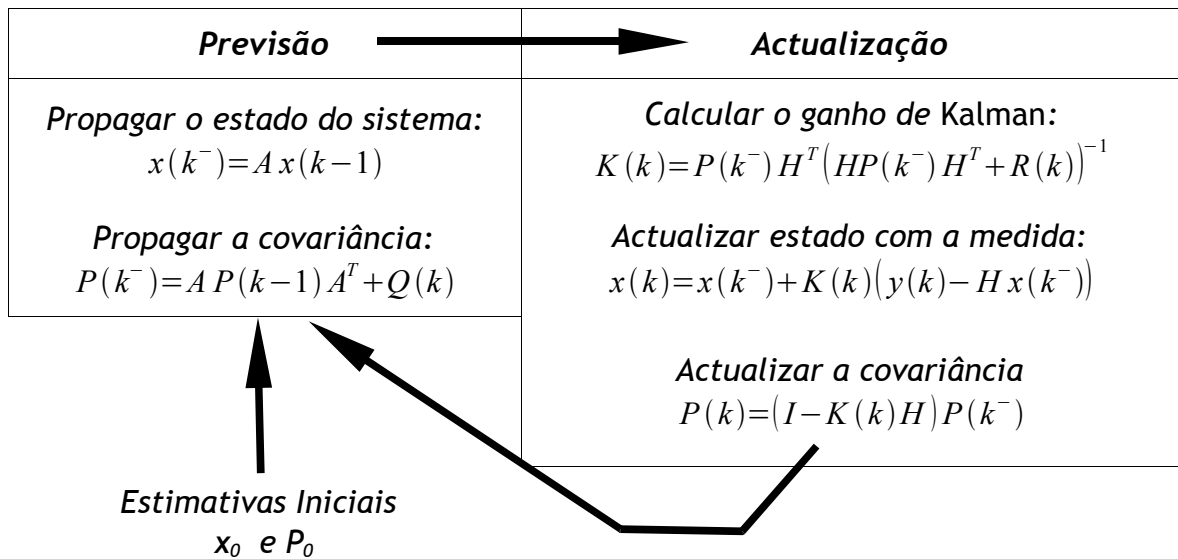
Se o Filtro de Kalman, considerar as medidas obtidas, então a inovação trazida pelas novas medidas é bastante considerada, $K(k) \approx H^{-1}$ provocando que o valor $P(k) < P(k^-)$. No caso de $K(k) = H^{-1}$ o valor máximo de decréscimo no valor da matriz $P(k)$ é conseguido.

No caso exactamente oposto, que em termos do ganho de Kalman este seja $K(k) \approx 0$, implica que a actualização terá um peso muito pequeno em relação à previsão, propiciando, que o valor da incerteza do estado se mantenha $P(k) = P(k^-)$.

O tempo que o Filtro de Kalman necessita, afim de voltar a se acreditar nas inovações propostas pelas medidas obtidas, depende do ruído presente na fase de previsão do algoritmo e, conseqüente menos da influência deste, na matriz da incerteza.

Na Tabela 3.1 é apresentado um resumo do algoritmo do Filtro de Kalman Linear.

Tabela 3.1: Resumo das operações do Filtro de Kalman Linear



Após esta apresentação e dedução do algoritmo, é possível concluir que a determinação do ganho de Kalman corresponde ao ponto nevrálgico de todo o processo de estimação realizado pelo Filtro de Kalman.

No ponto seguinte, irei provar que o Ganho de Kalman deriva do facto do Filtro de Kalman ser um estimador de variância mínima, ou seja, que minimiza a variância na estimação do estado.

3.8 Estimador de Variância Mínima

É possível provar, que a formulação do ganho de Kalman, utilizada no Filtro de Kalman para determinar que parte da inovação, deverá ser utilizada para melhorar a estimação do estado, minimizando a covariância do erro na estimação do próximo estado. Ou seja, quando uma nova medida é obtida e validada, o Filtro de Kalman utiliza o ganho de Kalman para incorporar a medida na estimação do estado, de tal modo, que a incerteza do estado estimado é mínima após a incorporação.

De modo a provar que o ganho de Kalman minimiza o erro de covariância a posteriori, provamos que $K(k)$, minimiza o somatório das variâncias presentes na matriz da incerteza à posteriori $P(k^+)$, é na realidade o $K(k)$ apresentado no algoritmo.

As variâncias na estimativa posteriori encontram-se na diagonal principal da matriz das incertezas. O traço da matriz corresponde ao somatório dos valores presentes na diagonal principal, se relacionarmos à incerteza a

posteriori $P(k^+)$ com o ganho de Kalman $K(k)$, então podemos derivar o traço de $P(k^+)$ em função a $K(k)$. De modo a encontrar $K(k)$, que minimiza o traço, igualamos a derivada a zero e resolvemos em relação a $K(k)$. [11]

1. Expressar a covariância à posteriori P_k^+ em termos de $K(k)$, utilizando para isso as equações 3.21, 3.22 e 3.46.

$$\begin{aligned} P_k^+ &= E[(e_k^+)(e_k^+)^T] \\ &= E[(x_k - \hat{x}_k^- - K(z_k - H\hat{x}_k^-))(x_k - \hat{x}_k^- - K(z_k - H\hat{x}_k^-))^T] \end{aligned} \quad (3.53)$$

2. Partindo do ponto anterior e incorporando o modelo das medidas 3.12, a definição do erro à priori 3.17 e ainda o pressuposto de média zero para o ruído do sistema

$$\begin{aligned} P_k^+ &= E[(x_k - \hat{x}_k^- - K_k(Hx_k + v_k - H\hat{x}_k^-))(x_k - \hat{x}_k^- - K_k(Hx_k + v_k - H\hat{x}_k^-))^T] \\ &= E[((I - K_k H)(x_k - \hat{x}_k^-) + (K_k v_k))((I - K_k H)(x_k - \hat{x}_k^-) + (K_k v_k))^T] \\ &= (I - K_k H) E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] (I - K_k H)^T \\ &\quad + K_k E[v_k v_k^T] K_k^T + 2(I - K_k H) E[(x_k - \hat{x}_k^-) v_k^T] K_k^T \\ &= (I - K_k H) P_k^- (I - K_k H)^T + K_k R_k K_k^T \end{aligned} \quad (3.5)$$

4)

3. Processando a derivada do traço do resultado acima e utilizando

$$\frac{\partial \text{tr}}{\partial A} (ABA^T) = 2AB \quad (3.55)$$

então

$$\begin{aligned} \frac{\partial \text{tr}(P_k^+)}{\partial K_k} &= \frac{\partial}{\partial K_k} [\text{tr}((I - K_k H) P_k^- (I - K_k H)^T + K_k R_k K_k^T)] \\ &= \frac{\partial}{\partial K_k} [\text{tr}((I - K_k H) P_k^- (I - K_k H)^T) + \text{tr}(K_k R_k K_k^T)] \\ &= -2(I - K_k H) P_k^- H^T + 2K_k R_k \\ &= -2P_k^- H^T + 2K_k (H P_k^- H^T + R_k) \end{aligned} \quad (3.56)$$

4. Igualando a derivada obtida a zero e resolvendo para $K(k)$

$$K_k = P_k^- H^T (H P_k^- H^T + R_k)^{-1} \quad (3.57)$$

A expressão $K(k)$ que acabamos de derivar, é exactamente igual à expressão utilizada no Filtro de Kalman para determinar o ganho de Kalman. Sendo assim, está provado, que esta expressão na realidade minimiza o erro da covariância a posteriori. [10]

3.9 Extensões ao Filtro de Kalman

3.9.1 Introdução

Vamos agora, neste sub-capítulo, apresentar algumas das extensões desenvolvidas a partir da definição do Filtro de Kalman.

As extensões foram desenvolvidas de modo a expandir o universo de aplicação do Filtros de Kalman, tanto para o nível discreto, com o Filtro de Kalman Discreto, como no nível não linear, com o Filtro de Kalman Extendido, entre muitos outros.

3.9.2 Filtro de Kalman Discreto

O Filtro de Kalman Discreto, como o nome indica, é utilizado em sistemas lineares definidos em tempo discreto, apresentando as mesmas, duas operações básicas: previsão e actualização.

Na operação de previsão é utilizado o estado anterior do sistema, ou então, a estimativa inicial, se estivermos na iteração inicial, sendo esta informação utilizada para prever o novo estado. Esse processamento realiza-se recorrendo ao modelo em espaço de estados do sistema. Nesta mesma operação realiza-se também, a previsão da covariância do estado, utilizando-se para isso a informação anterior da covariância do estado e o próprio modelo do sistema.

Na operação de actualização, no caso de existir medidas válidas, é fundida a informação presente das medidas adquiridas com o estado estimado na operação de previsão.

Considerando então as condições iniciais:

$$E(x(0)) = X_0 \quad cov(x(0)) = P_0 \quad (3.58)$$

E o modelo do sistema:

$$x(k) = \Phi x(k-1) + Gu(k-1) + w(k-1) \quad (3.59)$$

Com as observações:

$$y(k) = Hx(k) + v(k) \quad (3.60)$$

Propagação de estado será então:

$$x(k^-) = \Phi x(k-1) + Bu(k-1) \quad (3.61)$$

A associada covariância:

$$P(k^-) = \Phi P(k-1) \Phi^T + Q(k) \quad (3.62)$$

Para a actualização deve-se calcular o Ganho de Kalman:

$$K(k) = P(k^-) H^T (H P(k^-) H^T + R(k))^{-1} \quad (3.63)$$

E actualizar a covariância:

$$P(k) = (I - K(k)H) P(k^-) \quad (3.64)$$

Actualização do estado:

$$x(k) = x(k^-) + K(k)(y(k) - H x(k^-)) \quad (3.65)$$

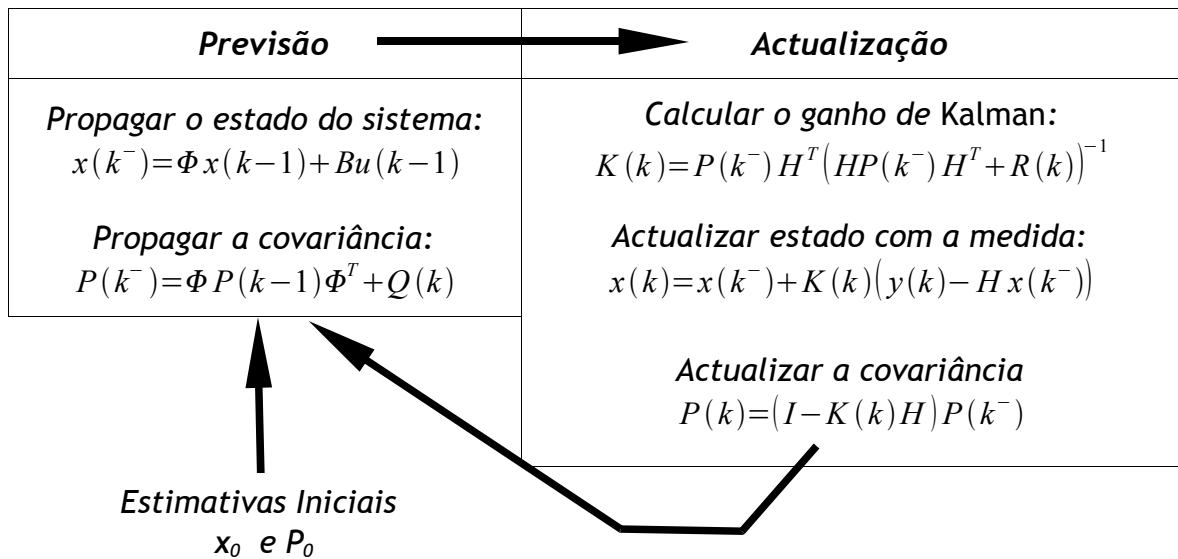
Por vezes e em vez da equação 3.58 pode utilizar-se:

$$P(k) = (I - K(k)H) P(k^-) (I - K(k)H)^T + K(k) R K(k)^T \quad (3.66)$$

A equação 3.61 é mais complexa mas é válida para qualquer ganho $K(k)$, mesmo que este seja não óptimo, o que pode ser interessante para algumas aplicações práticas.

Como resumo atente-se à Tabela 3.2 que ilustra o funcionamento do Filtro de Kalman Discreto.[2]

Tabela 3.2: Resumo das operações do Filtro de Kalman Discreto



3.9.3 Sistemas Dinâmicos Não Lineares

No mundo real, muitos dos sistemas dinâmicos e sensores não são lineares, mas também não são muito não lineares ou impossíveis de linearizar. Ou seja, é possível definir um sistema não linear, por um sistema linear com um certo erro.

Em vez de se assumir para o Filtro de Kalman, que todos os modelos como lineares, assumimos que o sistema e o modelo das medidas são não lineares.

Modelo Não Linear do Sistema

Se o sistema é não linear, então não pode ser definido como apresentado uma equação linear 3.5, mas sim, com uma equação não linear do tipo:

$$x_k = f(x_{k-1}) + w_{k-1} \quad (3.67)$$

em que $f()$ corresponde a função que representa o sistema não linear, relacionando-o com o estado anterior, com o estado actual do mesmo, e onde w_{k-1} representa novamente o ruído associado ao modelo do sistema.

De novo, é assumido que o ruído apresenta uma distribuição Gaussiana, sendo independente, considerado ruído branco e com média nula.

Modelo Não Linear das Medidas

Como para o caso do modelo do sistema, agora também o modelo das medidas não pode ser considerado linear e governado pela equação 3.6, passando a ser expresso por:

$$z_k = h(x_k) + v_k \quad (3.68)$$

onde $h()$ é a função não linear que relaciona o estado do sistema com as medidas obtidas, e onde v_k continua a representar o ruído associado a medida, e mantendo as características de distribuição Gaussiana, independente, com média nula e branco.

3.9.4 Filtro de Kalman Extendido

O Filtro de Kalman Extendido, como referido acima, é uma extensão do Filtro de Kalman original, que é utilizado no caso de sistema não lineares. Este Filtro funciona na premissa de que é possível linearizar a função de transferência do processo em torno do valor estimado, aplicando depois, as operações do Filtro de Kalman base, como se de um sistema linear se tratasse.

Com a linearização da função de transferência e a possível natureza não linear do sistema, não existem garantias teóricas de que a solução converja nem que seja ótima. Estas situações podem ser minimizadas com um projecto mais cuidadoso, pelo menos no problema da convergência do método.

Para que o método funcione e, como já foi apresentado no caso Discreto,

as condições iniciais são definidas:

$$E(x(0)) = X_0 \quad cov(x(0)) = P_0 \quad (3.69)$$

Neste caso o sistema estocástico não linear é definido pelas seguintes equações:

$$\begin{aligned} x(k) &= f(x(k-1), u(k), w(k-1)) \\ y(k) &= h(x(k), v(k)) \end{aligned} \quad (3.70)$$

Em que $w(k)$ e $v(k)$ representam as mesmas restrições sobre o método como explicitado na apresentação teórica e formulação do Filtro de Kalman, não se alterando com a definição da extensão para sistemas não lineares, ou seja, que ambos representam processos Gaussianos de média nula. Tendo isso em consideração, é possível validar a seguinte aproximação:

$$\begin{aligned} \tilde{x}(k) &= f(\hat{x}(k-1), u(k), 0) \\ \tilde{y}(k) &= h(\hat{x}(k), 0) \end{aligned} \quad (3.71)$$

Onde \hat{x} corresponde a estimativa à posteriori do estado, que é resultado da iteração anterior do Filtro de Kalman, ou então corresponde à estimativa inicial do estado.

Torna-se possível determinar a linearização do modelo do sistema tendo em consideração aos resultados de 3.69:

$$\begin{aligned} x(k) &\approx \tilde{x}(k) + A(x(k-1) - \hat{x}(k)) + W w(k-1) \\ y(k) &\approx \tilde{y}(k) + h(x(k) - \tilde{x}(k)) + V v(k) \end{aligned} \quad (3.72)$$

Para que a linearização seja possível, é necessário determinar a derivada do modelo e das componentes de ruído. Esta operação, é determinada através do jacobiano que passamos a apresentar:

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (3.73)$$

$$H = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \dots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_q}{\partial x_1} & \frac{\partial h_q}{\partial x_2} & \dots & \frac{\partial h_q}{\partial x_n} \end{bmatrix} \quad (3.74)$$

$$W = \frac{\partial f}{\partial w} = \begin{bmatrix} \frac{\partial f_1}{\partial w_1} & \frac{\partial f_1}{\partial w_2} & \dots & \frac{\partial f_1}{\partial w_n} \\ \frac{\partial f_2}{\partial w_1} & \frac{\partial f_2}{\partial w_2} & \dots & \frac{\partial f_2}{\partial w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial w_1} & \frac{\partial f_n}{\partial w_2} & \dots & \frac{\partial f_n}{\partial w_n} \end{bmatrix} \quad (3.75)$$

$$V = \frac{\partial h}{\partial v} = \begin{bmatrix} \frac{\partial h_1}{\partial v_1} & \frac{\partial h_1}{\partial v_2} & \dots & \frac{\partial h_1}{\partial v_n} \\ \frac{\partial h_2}{\partial v_1} & \frac{\partial h_2}{\partial v_2} & \dots & \frac{\partial h_2}{\partial v_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_q}{\partial v_1} & \frac{\partial h_q}{\partial v_2} & \dots & \frac{\partial h_q}{\partial v_n} \end{bmatrix} \quad (3.76)$$

De modo a simplificar a notação, não foi explicitada a dependência de A , W , H e V do instante de amostragem k .

Em termos de erro de predição e resíduo da medida temos:

$$\begin{aligned} \tilde{e}_x(k) &\equiv x(k) - \hat{x}(k) \\ \tilde{e}_y(k) &\equiv y(k) - \hat{y}(k) \end{aligned} \quad (3.77)$$

Podendo-se definir o erro como:

$$\begin{aligned} \tilde{e}_x(k) &\approx A(x(k) - \hat{x}(k-1)) + \epsilon(k) \\ \tilde{e}_y(k) &\approx H \tilde{e}_x(k) + \eta(k) \end{aligned} \quad (3.78)$$

em que $\epsilon(k)$ e $\eta(k)$ são variáveis aleatórias independentes com média nula e covariâncias WQW^T e VRV^T respectivamente, com Q e R a serem matrizes diagonais como definidas para o caso linear, representando igualmente o erro do modelo e da medida.

Com as semelhanças notórias entre as equações para o caso discreto 3.59 e 3.60 e as equações determinadas para este caso 3.70. Pode-se então

considerar um hipotético Filtro de Kalman para estas variáveis em que, a estimativa à posteriori do processo \hat{x}_k será:

$$\hat{x}(k) = \tilde{x}(k) + e(k) \quad (3.79)$$

com:

$$\begin{aligned} p(\tilde{e}_x(k)) &\sim N\left(0, E[\tilde{e}_x(k) \tilde{e}_x^T(k)]\right) \\ p(\epsilon_x(k)) &\sim N\left(0, E[W Q(k) W^T]\right) \\ p(\eta_x(k)) &\sim N\left(0, E[V R(k) V^T]\right) \end{aligned}$$

Fazendo que o valor esperado de $\hat{e}(k)$ seja 0 vem:

$$\hat{e}(k) = K_k \tilde{e}_y(k) \quad (3.80)$$

o que resulta em:

$$\hat{x}(k) = \tilde{x}(k) + K(k) \tilde{e}_y(k) = \tilde{x}(k) + K(k)(y(k) - \tilde{y}(k)) \quad (3.81)$$

Que mais não são do que as equações da actualização para o Filtro de Kalman Extendido.

Em termos de passos de execução para a realização do algoritmo do Filtro de Kalman Extendido temos que:

A propagação do estado será:

$$\hat{x}(k^-) = f(\hat{x}(k-1), u(k), 0)$$

A propagação da covariância:

$$P(k^-) = A(k)P(k-1)A(k)^T + W(k)Q(k-1)W^T(k) \quad (3.82)$$

O cálculo do ganho de Kalman $K(k)$:

$$K(k) = P(k^-)H(k)^T(H(k)P(k^-)H(k)^T + V(k)R(k)V(k)^T)^{-1} \quad (3.83)$$

Actualização da covariância:

$$P(k) = (I - K(k)H(k))P(k^-) \quad (3.84)$$

E finalmente a actualização do estado:

$$x(k) = x(k^-) + K(k)(y(k) - h(x(k^-), 0)) \quad (3.85)$$

O Filtro de Kalman Extendido utiliza uma linearização da dinâmica pelo que, mesmo que a estimativa inicial do estado $x(0)$ seja gaussiana, não há garantias que assim se mantenha ao longo do tempo. Em termos da propagação da covariância esta também é feita com recurso à linearização em torno do estado estimado. A conjugação destes dois factores de imprecisão são tanto mais importantes, quanto menos válida for a aproximação utilizada, daí a

necessidade de um projecto mais cuidado aquando do uso da versão extendida do Filtro de Kalman.

Na discussão acima, considera-se que o Filtro de Kalman Extendido utiliza a matriz A do processo dinâmico discretizado.

Como procuramos uma aproximação da dinâmica do sistema em tempo discreto, parte-se da seguinte descrição dinâmica do sistema no tempo contínuo:

$$\frac{dX(t)}{dt} = f(X(t), u(t_k), t), \quad t \in]t_{k-1}, t_k] \quad (3.86)$$

Calcula-se de seguida:

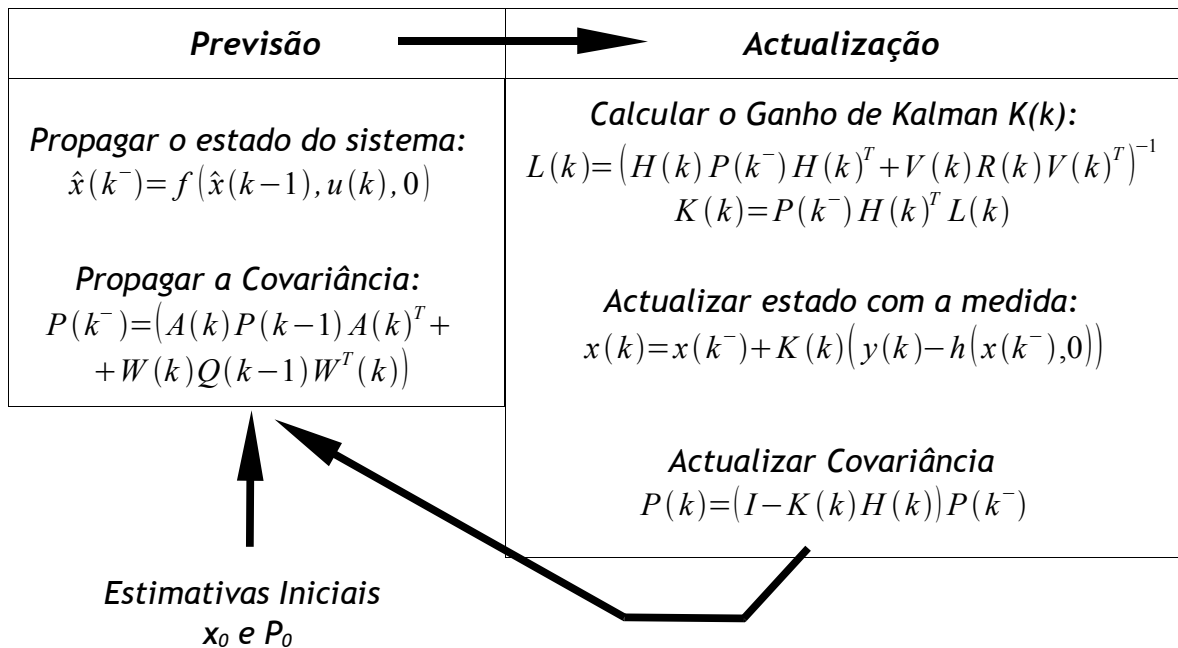
$$A_k = \left. \frac{\partial f}{\partial x} \right|_{\substack{x=x(t_k) \\ u=u(t_k) \\ t=t_k}} \quad (3.87)$$

E considerando os valores constantes entre instantes de amostragem obtemos finalmente:

$$A(k) = \exp(A_k(t_k - t_{k-1})) \quad (3.88)$$

Na Tabela 3.3 está resumido os passos para a realização do Filtro de Kalman Extendido.

Tabela 3.3: Resumo das operações do Filtro de Kalman Extendido



É necessário reforçar a ideia, que o jacobiano das matrizes $A(k)$ e $H(k)$ terão que ser avaliados para o estado estimado mais recente. Como o estado estimado não é conhecido à priori, o jacobiano não pode ser feito “off-line”, o que acrescentará peso ao processamento do Filtro. Este facto, não é de todo inconveniente, pois esta situação provoca, que o algoritmo apresente estimativas mais precisas e correctas, já que utiliza sempre a informação mais recente.

É de notar também, que no caso do Filtro de Kalman Extendido, as equações nas fases de previsão e actualização da incerteza, dependem do estado anterior do sistema. Sendo assim, a incerteza e o próprio ganho de Kalman, não podem ser analisados sem a informação presente do estado anterior do sistema. Algo que tanto no caso discreto como, no caso linear era possível realizar.

O Filtro de Kalman Extendido, tem se mostrado eficiente em muitas aplicações não lineares.[7] [8]

Em particular, o Filtro de Kalman Extendido tem sido utilizado com sucesso em aplicações em que o modelo do sistema é bem conhecido e linear, mas também em casos onde mesmo alguns dos parâmetros do sistema ou do modelo, não são conhecidos. Tratando-se dos parâmetros desconhecidos, e necessário acrescentar mais variáveis de estado, tornando o problema não linear. Estes parâmetros obtidos e estimados “online”.

3.9.5 Filtro de Kalman Extendido Iterativo

O Filtro de Kalman Extendido lineariza o modelo do sistema e o modelo das medidas, que não são lineares, redefinindo-os em cada instante de tempo através da última estimação para o estado.

Quando estamos perante não linearidades mais significativas, pode ser benéfico realizar iterações sobre os modelos obtidos, utilizando essa informação e redefinição para determinar um novo modelo.

A principal ideia por detrás do Filtro de Kalman Extendido Iterativo é de usar toda a informação das medidas de uma forma repetida, de modo a ajustar cada vez melhor, a estimativa do estado.

Podemos concluir, que esta acção pode ser benéfica, especialmente se nos debruçarmos no modo como o Filtro de Kalman Extendido incorpora as novas medidas. Se temos uma medida para incorporar, pretendemos ter a melhor estimativa possível do sistema. No instante de tempo k existe uma medida para ser incorporada, então neste caso a melhor estimativa para o estado do sistema, nesse instante, é \hat{x}_k^- . No entanto, no instante em que o Filtro de Kalman Extendido o processa, essa estimativa já não é a melhor estimativa para esse instante, mas sim a nova estimativa à posteriori para o sistema. Como devemos sempre utilizar a melhor estimativa para cada instante, devemos

utilizar a estimativa à posteriori em vez da estimativa à priori.

O Filtro de Kalman Extendido Iterativo, de uma forma iterativa redefine os modelos de modo a descobrir a melhor estimativa para o estado de modo a usa-la para incluir as novas medidas obtidas.

O Filtro consegue este funcionamento, processando o ganho de Kalman K_k com uma estimativa do estado à posteriori intermédia \hat{x}_k^i , onde i corresponde ao número da iteração. No instante zero $\hat{x}_k^0 = \hat{x}_k^-$, depois processando-se os restantes do modo apresentado em (3.89)

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ \hat{x}_k^{i+1} &= \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^i) - H_k (\hat{x}_k^- - \hat{x}_k^i)) \end{aligned} \quad (3.89)$$

Em termos do jacobiano da matriz H_k , ela é sempre avaliada para o estado estimado intermédio mais recente \hat{x}_k^i .

Em termos de limites de iterações, normalmente são aplicados dois métodos, ou um número fixo de iterações, ou então quando a diferença entre \hat{x}_k^{i-1} e \hat{x}_k^i passa abaixo de um certo “threshold”. Quando a fase iterativa do método termina, o Filtro de Kalman Extendido Iterativo processa a incerteza do estado estimado P_k^+ , utilizando o último estado estimado \hat{x}_k^i e as últimas matrizes K_k e H_k .

É de notar que o Filtro de Kalman Extendido Iterativo, processa estimativas de estado intermédias, mas não incertezas intermédias, realizando este processo, uma única vez e utilizando somente os dados mais actualizados, ou seja, os processados na última iteração, correspondendo à estimativa do estado mais precisa.

Mesmo sabendo, que o preço computacional do Filtro de Kalman Extendido Iterativo é muito superior ao Filtro de Kalman Extendido, a estimativa à posteriori do estado será melhor devido ter sido i vezes re-avaliada. Em termos das estimações futuras estas serão melhores, pois os modelos são mais precisos.[8]

3.10 Conclusões

O Filtro de Kalman é um algoritmo recursivo de fusão de informação, que tem sido utilizado com sucesso em inúmeras aplicações, desde decisão e previsão de preços de matérias primas, estimativa da posição de robots até controlo de naves espaciais. O Filtro de Kalman estima o estado de um sistema ruído utilizando medidas ruidosas, calculando a distribuição da probabilidade condicionada ao longo do tempo, dada as medidas obtidas e as incertezas associadas.

O Filtro de Kalman, pressupõem características definidas para os modelos e variáveis que utiliza, assumindo que todas elas são lineares, com ruídos independente e branco, apresentando uma distribuição Gaussiana com média nula, para além disso, assume também, que o estado inicial do sistema apresenta uma distribuição Gaussiana e é independente.

O algoritmo utilizado pelo Filtro de Kalman encontra-se dividido em duas fases. A primeira, intitulada de previsão, realiza a estimação do estado do sistema, baseado no modelo dinâmico do mesmo, determinando a estimativa e incerteza do sistema à priori. Na segunda fase, intitulada de actualização, processa-se a actualização da estimativa à priori com as medidas obtidas e validadas do sistema. Para isso, é determinada a inovação, que corresponde a diferença entre a medida estimada a partir da posição estimada à priori, e a real medida obtida. Este factor de inovação é depois incorporado na estimativa do estado à posteriori em função do ganho de Kalman obtido. É possível provar que o ganho de Kalman é determinado, de tal forma que minimiza o erro da estimativa à posteriori, o que define o Filtro de Kalman como um estimador de variância mínima.

Com o sucesso da aplicação do Filtro de Kalman, novas extensões foram desenvolvidas, tanto para sistemas discretos como para sistemas não lineares.

O Filtro de Kalman Extendido, apresenta-se como um algoritmo possível para o processamento e estimação de sistemas não lineares. Este realiza uma linearização do modelo do sistema e o modelo das medidas em torno do ponto de funcionamento, ou seja do estado estimado para o sistema.

Uma característica importante dos Filtros de Kalman e do Filtro de Kalman Extendido, em particular, é que o jacobiano H_k da equação 3.73, que calcula o ganho de Kalman, propaga correctamente a informação que a medida traz acerca de cada elemento do estado. Desta forma, se a medida não afectar todos os elementos do estado, as covariâncias do erro do estado são afectadas de acordo com a informação que a medida traz.

O custo computacional superior inerente a este algoritmo, devido ao processamento “online” de jacobianos é compensado pelo facto de incorporar o estado anterior e as medidas obtidas em modelos actualizados, o que proporciona estimações mais precisas.

É possível reduzir os erros que ocorrem na utilização do Filtro de Kalman

Extendido em sistemas muito não lineares, iterando a fase de actualização do Filtro de Kalman. A ideia encontra-se implementada, no intitulado Filtro de Kalman Extendido Iterativo, onde a estimativa do estado é incorporada sempre sobre a melhor estimativa presente. O Filtro de Kalman Extendido Iterativo, processa iterativamente estimativas intermédias para o estado, até que nota que não existe melhorias significativas.

O Filtro de Kalman e Filtro de Kalman Extendido só lidam com um conjunto de medidas sempre válidas, pois a fase de actualização necessita delas para ser realizada. No entanto, há possibilidade de contornar a inexistência de medidas de três formas: introduzindo covariâncias grandes nas medidas quando estas não se encontram disponíveis; repetir a actualização para as diferentes medidas que existam de facto ou então, simplesmente, não realizando a fase de actualização, considerando a estimativa realizada na previsão como a estimativa do filtro.

Todas as hipóteses não são perfeitas, a primeira pode provocar problemas numéricos, a segunda corresponde a uma aproximação à teoria, mas utilizada na prática, a terceira é a mais leve de todas mas, se o número de medidas indisponíveis for elevada, poderá começar a adulterar em demasia o funcionamento do método, passando a ser simplesmente um processo baseado unicamente no modelo e não na fusão de informação.

Se o conjunto das medidas não trazer (nunca) informação acerca de todas as partes do vector de estado, então o filtro torna-se divergente visto que o processo não é observável.

Capítulo 4

Trabalho Desenvolvido

4.1 O Robot de Limpeza

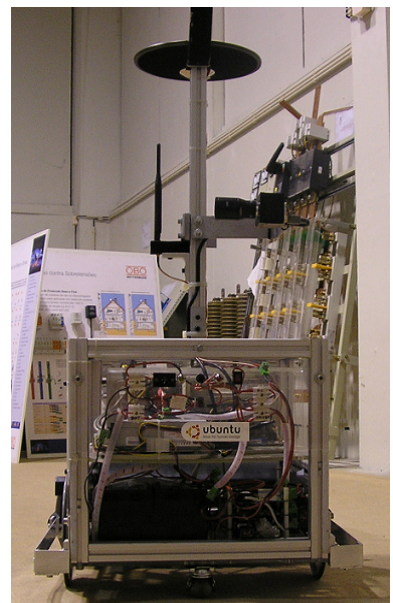


Figura 4.1: Imagens Actuais do Robot de Limpeza

4.1.1 Apresentação Geral do Robot de Limpeza

O robot de limpeza apresenta uma estrutura em alumínio, cúbica, com uma configuração de locomoção do tipo diferencial.

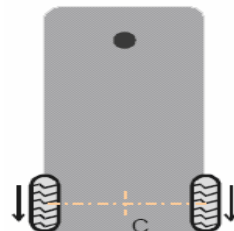


Figura 4.2: Robot com Locomoção Diferencial

A estrutura física do robot, já se encontrava previamente montada, mas durante o projecto foi sujeita a diversas alterações.

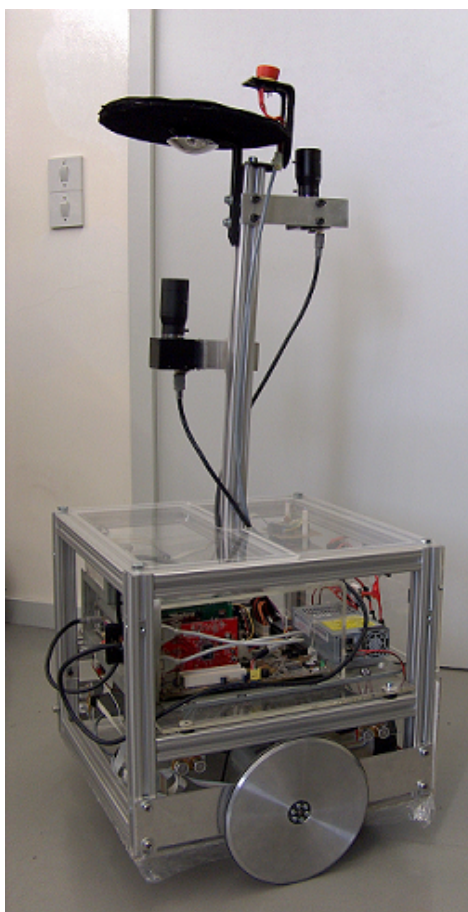


Figura 4.3: O Antes e o Depois do Robot de Limpeza

Como é possível verificar pela figura 4.3, o robot foi bastante alterado.

Tendo hoje, uma estrutura com uma aparência exterior semelhante, mas bastante retocada, sendo agora, muito mais fiável, robusto, modelar e acima de tudo, encontra-se bem documentado.

Na figura 4.3 é possível perceber o porquê do esquema de locomoção apresentado na figura 4.2. Inicialmente o robot tinha uma forma cúbica, mas agora com as suas novas capacidades e funções operacionais, recebeu alguns “upgrades” sendo um deles bastante visível e representado pelo aspirador acoplado à frente do robot.

Em relação às cameras firewire que o robot inicialmente apresentava, uma foi prescindida e em relação à outra camera, foram definidas novas funções o que levou que a sua posição relativa no robot, assim como, a sua orientação, tivessem que ser alteradas.

Os sonares foram recolocados na estrutura, agora numa posição mais elevada; problemas de interferência entre sonares e entre o chão e os sonares foram minimizados e o alcance máximo aumentado. Estes efeitos foram conseguidos devido a um misto entre a recolocação dos mesmos na estrutura, o desenvolvimento de um novo código para os drivers de controlo do sonares, do barramento I2C, assim como da definição de novas estratégias de disparo.

Um novo tipo de sensor foi adicionado, Sharps. Estes sensores apresentam um tipo de medida semelhante à dos sonares, mas com a grande vantagem de serem muito mais direccionais e precisos.

Por fim, um sistema de controlo e supervisão desenvolvido de raiz, com uma arquitectura de software modular, com a capacidade de facilmente se integrar no processamento de novos sensores, algoritmos de auto-localização, funções de movimentação e também a capacidade de gerar um histórico de todas as acções e estados do robot, podendo-se assim depois, visitar “offline” a missão realizada e analisar todos os dados obtidos, acções tomadas e diagnosticar problemas no funcionamento.

Arquitectura Tecnológica

O robot de limpeza apresenta uma arquitectura tecnológica complexa. Integrando diversos dispositivos externos, com diferentes necessidades de comunicação, assim como uma ligação à “internet”, que proporciona acções de monitorização, supervisão, ou mesmo de alteração de código e definição de novas missões.

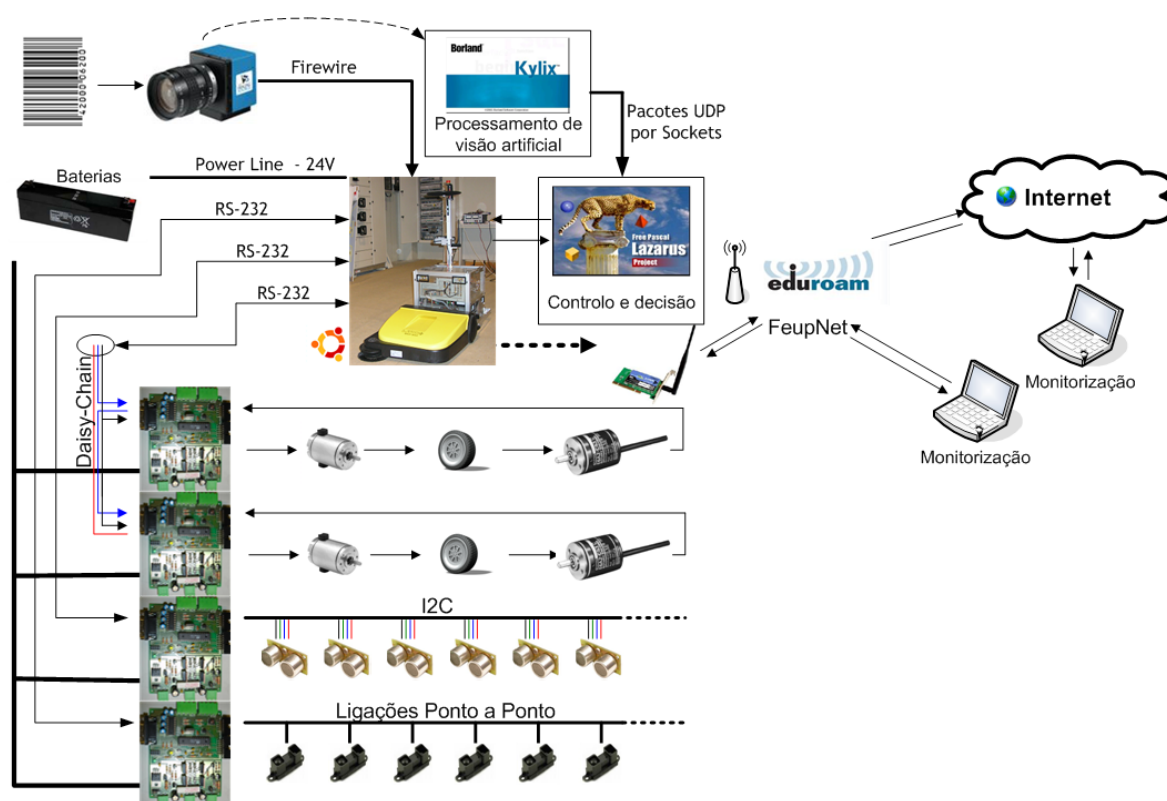


Figura 4.4: Arquitectura Tecnológica do Robot de Limpeza

É possível visualizar na figura 4.4 os diferentes tipos de comunicação utilizados, para a integração total de todos os sistemas no robot de limpeza.

Sejam eles sobre firewire, como entre a camera e o robot, ou por socket com pacotes UDP entre as aplicações de localização de códigos de barras e o programa de controlo e decisão do Robot de Limpeza, ou ainda através de uma comunicação série RS-232, aplicando uma lógica de “daisychain” de modo a interligar mais do que um dispositivo através da mesma conexão física, estratégia utilizada entre os programas de controlo drivers dos motores, ou mesmo numa ligação serie, mas do tipo “unicast”, estratégia utilizada para as outras duas placas de drivers.

De modo a se conseguir testar e interligar todos este subsistemas, foram desenvolvidas diversas ferramentas computacionais de auxilio ao projecto e seu progresso. A maioria das ferramentas foram desenvolvidas em “Lazarus” versão 0.9.24, utilizando o compilador de “FreePascal” versão 2.2.1, tudo sobre um

sistema operativo Linux.

As principais aplicações desenvolvidas foram:

- **CleanRob I2C Tester Program** - ferramenta de teste, despiste e configuração das comunicações com o driver dedicado à comunicação via protocolo I2C. O código de firmware presente no driver foi também desenvolvido no seio da equipa do Robot de Limpeza;
- **CleanRob Motor Tester Program** - ferramenta de teste, despiste, configuração e monitorização da comunicação com os drivers que controlam os motores. Esta ferramenta é também capaz de traçar gráficos da evolução da corrente e velocidade em cada um dos motores para além de criar um ficheiro de texto, com informação dos motores, que mais tarde pode ser utilizada para recalcular os parâmetros dos PID's;
- **CleanRob Map Creator** - ferramenta de construção ou reconstrução de ficheiros de mapa. Este tipo de ficheiros são utilizados por algumas das aplicações desenvolvida, para representação do “mundo” onde o Robot de Limpeza se encontra;
- **CleanRob Control Program** - aplicação fulcral para todo o funcionamento do projecto, esta aplicações contempla toda a “inteligência” do Robot de Limpeza, sendo responsável por o colocar a realizar as suas acções de sensorização, movimentação e de localização. Esta aplicação gera um ficheiro com a informação obtida e processada em cada ciclo de controlo, podendo este ficheiro depois ser lido pela aplicação “CleanRob Log Replay Program” onde de um modo “offline” se pode analisar as movimentações e estimações realizadas pelo Robot de Limpeza;
- **CleanRob Log Replay Program** - principal aplicação de análise e tratamento de dados do projecto do Robot de Limpeza. Esta ferramenta é responsável pelo replay do ficheiro gerado pelo “CleanRob Control Program”, sendo capaz de gerar gráficos, adiantar ou atrasar o instante de tempo da análise, etc., em suma, é uma ferramenta complexa de análise e tratamento de dados reais obtidos do Robot de Limpeza.

Existe ainda uma outra aplicação, desenvolvida pelo Professor Doutor Paulo José Cerqueira Gomes da Costa, um dos orientadores desta tese:

- **Bar Code Localization** - aplicação de localização de códigos de barras do tipo “*Interleaved 2 from 5*”. Esta aplicação é também capaz de determinar a distância ao código de barras, em termos cartesianos, e ainda o ângulo entre a posição do código de barras e a camera. Esta informação é então enviada por socket, num pacote UDP assim como o número do código de barras e o número de linhas visualizadas.

Foram também desenvolvidas inúmeras folhas de cálculo, scripts Matlab e outros programas mais pequenos e simples, sempre numa lógica de auxílio ou

tratamento de dados relacionados com o projecto do Robot de Limpeza.

Sistema de Controlo de Motores e Locomoção

O controlo e monitorização do estado do sistema de movimentação é realizado pelo programa de controlo do robot, através de uma comunicação RS-232 sobre o protocolo “DaisyChain”, para dois drivers de potência que se encontram numa plataforma próximas dos motores.

As funções principais dos drivers são:

- recepção de referências de velocidade provenientes do sistema de controlo;
- recepção dos dados provenientes dos encoders, que se encontram ligados às rodas do robot;
- alimentação dos motores;
- controlo da velocidade de cada uma das rodas do robot através de um controlador PID;
- envio de informação sobre o estado actual das medidas dos encoders, correntes absorvidas pelos motores, etc.

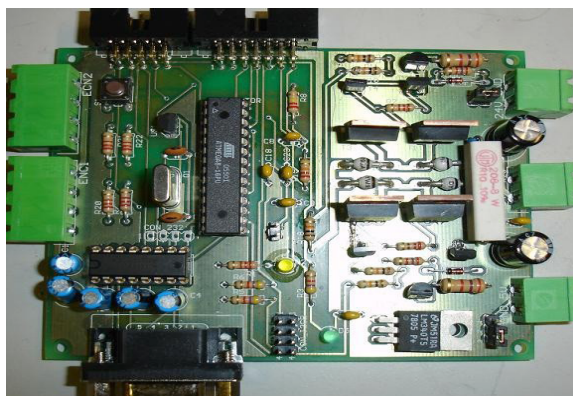


Figura 4.5: Fotografia dos Drivers utilizados para o controlo e alimentação dos motores do Robot de Limpeza

Ligada a cada uma das rodas do robot, encontra-se um encoder, modelo E6A2-CW5C da Omron, responsável por medir o deslocamento realizado.

A informação obtida é então utilizada como medida de comparação no processamento do controlador PID, presente no drivers, e ainda, é enviada para o programa de controlo, afim de este processar o deslocamento realizado pelo robot e estimar a velocidade actual do mesmo.

Tendo noção das medidas conseguidas com a informação proveniente dos encoders e conhecendo o tipo de locomoção do robot, descrita na figura 4.2, podemos converter as velocidade individuais por roda, para as suas componentes

tangencial e angular da velocidade do robot:

$$V_t = V_r = \frac{V_1 + V_2}{2} \quad (4.1)$$

$$\omega_r = \frac{V_1 - V_2}{D} \quad (4.2)$$

Com esta informação e com o conhecimento da posição anterior do robot, é possível determinar a estimativa da sua posição baseada na odometria, necessitando primeiro, de descobrir as velocidades segundo os eixos coordenados e determinar a direcção actual do robot.

$$\theta_k = \theta_{k-1} + \omega_r \cdot \Delta t \quad (4.3)$$

$$V_x = V_t \cdot \cos(\theta_k) \quad (4.4)$$

$$V_y = V_t \cdot \sin(\theta_k) \quad (4.5)$$

Podendo agora descobrir uma estimativa para posição, em termos cartesianos, do robot:

$$x_k = x_{k-1} + V_x \cdot \Delta t \quad (4.6)$$

$$y_k = y_{k-1} + V_y \cdot \Delta t \quad (4.7)$$

Leitor de Códigos de Barras

As áreas de acção de trabalho do robot de limpeza apresentam, dispostas nas suas paredes, códigos de barras do tipo “*Interleaved 2 from 5*” como representado na figura 4.6.

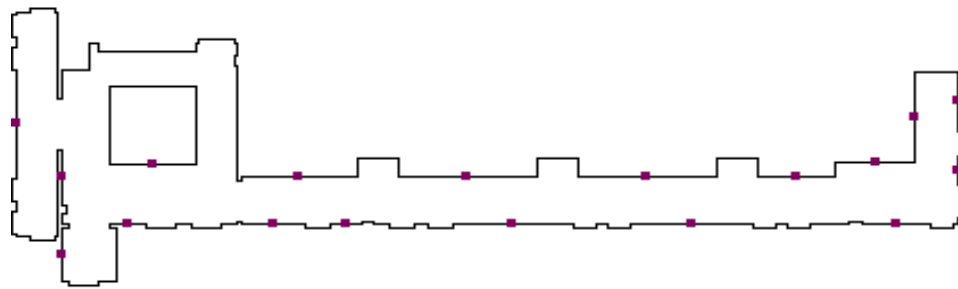


Figura 4.6: Mapa do Segundo Piso do Edifício Poente do Departamento de Engenharia Electrotécnica da Universidade do Porto, com a disposição dos códigos barras pela área

É assumido que todas as posições dos códigos de barras no “mundo” se encontram bem definidas e são conhecidas, tanto em termos de posição cartesiana (x_{CB}, y_{CB}) para além da orientação dos mesmos θ_{CB} .

A distância ao código de barras é dada, por uma aplicação que funciona em paralelo com a aplicação de controlo do robot, criada por um orientadores do projecto, Professor Doutor Paulo José Cerqueira Gomes da Costa.

Esta aplicação, é capaz de identificar e medir distância nas suas três componente cartesianas e ainda a perspectiva, baseando-se na imagem captada pela camera firewire.



Figura 4.7: Imagem do Programa de Medição de Distâncias a Códigos de Barras

A aplicação envia informação via socket através de um pacote UDP, com as medidas relativas da distâncias, número de linhas avistadas e ainda o número do código de barras visualizado.

Sonares

No início do projecto, o robot estava equipado com seis sonares, para detecção de distâncias a obstáculos.



Figura 4.8: Sonares SRF08

Os sonares do modelo SRF08, figura 4.8, comunicam através do protocolo de comunicação I2C, por um barramento, que interliga todos os sonares do robot, permitindo uma transmissão série com 8 bits e uma velocidade de transferência de 400 Kbits/s.

Dado que a característica de comunicação do protocolo I2C é um meio de comunicação partilhado, é necessário definir para cada sensor, um endereço, servindo este como identificação, conseguindo assim o algoritmo de comunicação do sonar capturar a mensagem destinada e responder ao pedido da mensagem.

A detecção de obstáculos por parte do sonar depende, exclusivamente, da reflexão do som no obstáculo, ao contrário de outros sensores, nomeadamente ópticos. Este sonar, não tem como característica, um foco direcional da onda, pelo que a difusão do eco tem um ângulo de abertura aproximadamente de 30°, como representado na figura 4.9.

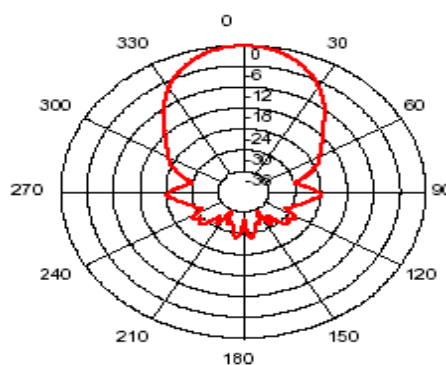


Figura 4.9: Difusão Teórica para o Eco do Sonar

Sharps

O robot de limpeza encontra-se equipado actualmente com quatro sharps, ou seja, quatro dispositivos de medição de distâncias por infra-vermelhos.



Figura 4.10: Dispositivo de medição de distâncias por infra-vermelhos (Sharp GP2Y0A02YK0F)

A gama de Sharp que actualmente equipa o Robot de Limpeza, apresenta uma gama de medição entre 20cm e 150cm.

Em termos do tipo de medida e informação conseguida através destes dispositivos, é muito semelhante à dos sonares mas, apresentam uma grande vantagem sobre os anteriores, já que são muito mais direccionais, sendo a desvantagem a gama de medidas que é muito inferior, em teoria, a dos sonares.

Os sharps são sensores analógicos, o que promoveu o desenvolvimento e implementação de um novo driver, responsável pela interligação entre os sharps e o Robot de Limpeza. A placa desenvolvida é baseada numa PIC16F882 comunicando via interface RS-232 e contém 11 conversores analógico digitais, o que irá facilitar as expansões futuras em termos de número de sharps na estrutura.

4.2 Calibração dos PID's

4.2.1 Abordagem Implementada

De modo a controlar verdadeiramente os motores e especialmente a sua resposta ao degrau, foi necessário calibrar os controladores PID desenvolvidos em software e integrados nos microcontroladores presentes nas placas de circuito impresso dos drivers de potência.

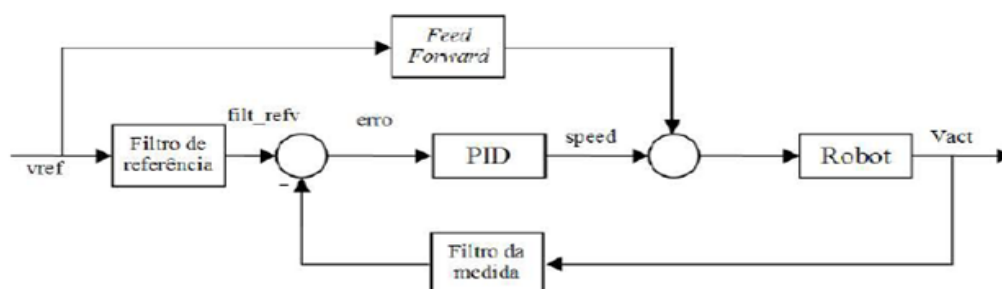


Figura 4.11: Esquema do Controlador PID implementado nos drivers de controle dos Motores

Na realidade, o controlador implementado no microcontrolador não é um simples controlador PID, mas contém também um filtro de medida, um filtro de referência e ainda uma malha de feed-forward.

Devido a existir uma camada de software, mais uma camada de controlo, necessitando ambas de ser calibradas, optou-se por utilizar somente o controlador de PID, uma vez que, para a aplicação pretendida as outras afinações não trouxeram mais valias significativas.

Para a determinação dos parâmetros do PID foi utilizada uma abordagem do tipo controlo por modelo interno.

Esta decisão foi realizada devido à planta do sistema não ser completamente conhecida, devido à facilidade de determinação dos valores dos parâmetros para o PID e apresentar um único factor de ajuste, que se encontra directamente relacionado com o tempo de resposta do sistema.

Foi também escolhido o modelo de primeira ordem com atraso.

A opção de sistema com atraso é justificada pelo facto de existir uma diferença entre o instante em que a informação do robot é obtida, e o instante em que a informação é processada pelo sistema de controlo, já que, o programa realiza um ciclo de controlo cada 100ms.

Em termos de robot este tratar-se de um sistema de primeira ordem, já que é assumido, que o robot é capaz de realizar acelerações constantes, e com o sistema dinâmico considerado é em termos de velocidade, então estamos perante um sistema de primeira ordem.

Tabela 4.1: Modelo de Primeira Ordem com Atraso, definido para a calibração dos parâmetros do PID pelo Método de Chien

Modelo do Processo	Parâmetros do Controlador		
	$k_c \cdot k$	τ_i	τ_D
$\frac{k_p \cdot e^{-Ls}}{1 + \tau s}$	$\frac{\tau}{\tau_{CL} + L}$	τ	

O método de Chien define, para este tipo de modelo, que o melhor controlador será do tipo PI, podendo os parâmetros determinados ser apresentados na Tabela 4.1.

O factor τ_{CL} corresponde à medida que controla a relação entre o tempo de resposta em malha aberta e a pretendida, em malha fechada. Foi assumido que, se pretendia um tempo de resposta que poderia ser até ao dobro do conseguido em malha aberta. Sendo assim, definiu-se $\tau_{CL}=2$

Com a calibração do PID pretende-se diminuir o erro em regime permanente, assim como, definir um melhor tempo de resposta do sistema.

Para a determinação dos dados sobre os motores, de modo a realizar a calibração, foi utilizada a seguinte abordagem: colocou-se os motores em funcionamento em malha aberta, aplicando quatro degraus de velocidade:

1. velocidade no sentido positivo, durante aproximadamente 4 segundos, com um valor de 280 ticks/amostragem;
2. rampa de desaceleração até uma velocidade de referência de 0 ticks/amostragem, permanecendo nesta fase durante 3 segundo;
3. velocidade no sentido negativo, durante aproximadamente 4 segundos, com um valor de -280 ticks/amostragem;
4. rampa de desaceleração até uma velocidade de referência de 0 ticks/amostragem, permanecendo nesta fase durante 3 segundo.

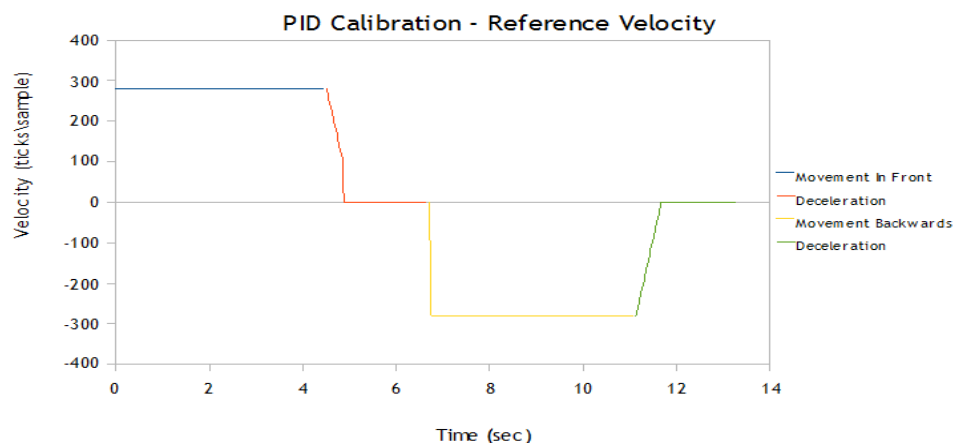


Figura 4.12: Referências de Velocidade Aplicadas na determinação dos Parâmetros do PID

4.2.2 Dados Experimentais

Na página web da dissertação, estão apresentados os dados em bruto de cada um dos “runs” de calibração.

Estes dados foram obtidos directamente, por uma aplicação realizada especificamente para o Robot de Limpeza, chamada de “Motor Tester” apresentada no início deste capítulo. O processamento dos dados é realizado com auxílio de uma “template” da aplicação Calc do OpenOffice, onde colando as informações escritas pela aplicação “Motor Tester”, esta realiza, de um forma automática, todos os cálculos necessários para determinar os parâmetros do PID, preenchendo diversas tabelas e desenhando gráficos auxiliares, que ajudam na análise de resultados.

Todos os dados apresentados abaixo são fruto da conjugação desta duas ferramentas, realizadas no decurso deste projecto.

De seguida, apresentarei os dados já tratados, para cada um dos motores, através de representações gráficas dos “runs” realizados e ainda dos valores calculados para cada um dos parâmetros do controlador, sendo desta vez apresentados em forma tabular.

O atraso do sistema é de 100ms , correspondendo este valor ao período entre chamadas do ciclo de controlo no programa que supervisiona e controla todo o processamento do robot.

O tempo de estabelecimento, corresponde ao tempo decorrente desde o instante em que o sistema começa a responder à solicitação de movimento e o instante em que o valor de regime permanente é obtido.

As referências enviadas para os drivers, provenientes do programa de controlo, são em termos de velocidade, assim sendo, é sobre os dados presentes nos gráficos apresentados nas figuras 4.13 e 4.17 que se calculam os parâmetros do PID para os motores.

Motor da Direita

Analisemos agora os dados referente ao motor da direita.

Inicialmente, este irá movimentar se no sentido positivo considerado para a movimentação do robot, sendo então a velocidade medida positiva, passando depois num segundo momento a velocidade a ser negativa.

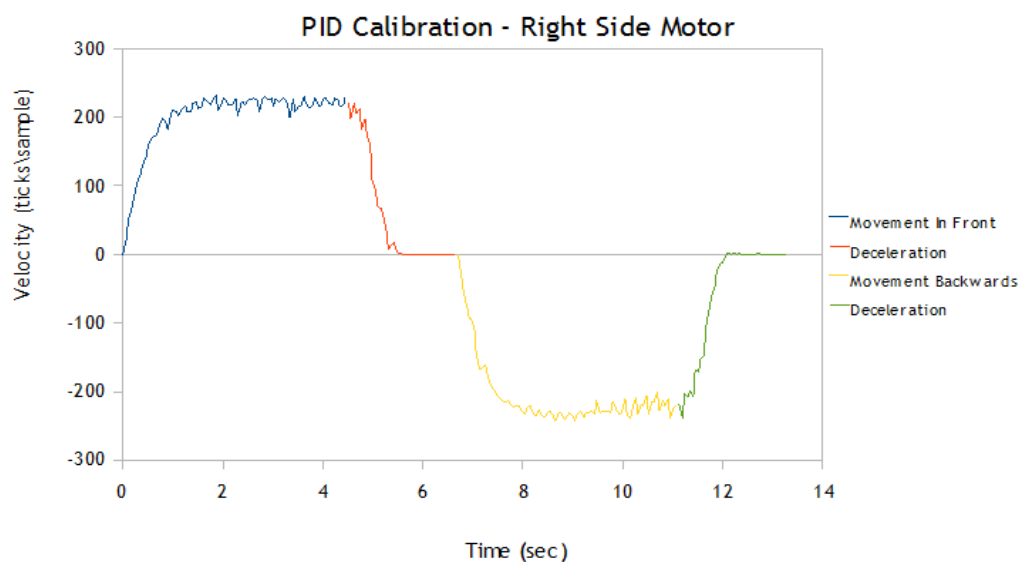


Figura 4.13: Gráfico da Resposta em Malha Aberta do Motor da Direita, em termos de Velocidade

É possível resumir os dados experimentais, obtidos para o motor da direita e necessários para a calibração do controlador em forma de tabela, sendo apresentados na Tabela 4.2.

Tabela 4.2: Resumo dos Principais Dados obtidos sobre a Dinâmica do Motor da Direita

<i>Características de Dinâmica para o Motor da Direita</i>	<i>Movimentação no Sentido Positivo</i>	<i>Movimentação no Sentido Negativo</i>
<i>Atraso (seg)</i>	0,1000	0,1000
<i>Valor Médio em Regime Permanente (ticks/amost)</i>	216,3683	-227,0039
<i>Tempo de Estabelecimento (seg)</i>	0,9496	0,7542

Tendo em consideração os dados apresentados na Tabela 4.2 e fórmulas de cálculo dos parâmetros de configuração do PID, segundo o Método Chien, presente na Tabela 4.1, podemos determinar os parâmetros a utilizar para o Motor da Direita, que são apresentados na Tabela 4.3.

Tabela 4.3: Resumo dos Dados obtidos para a Calibração do Controlador PID, segundo o Método de Chien, para o Motor da Direita.

<i>Método de Chien</i>	<i>Movimentação no Sentido Positivo</i>	<i>Movimentação no Sentido Negativo</i>
<i>Coeficiente de Chien</i>	2,0000	2,0000
<i>Ganho da Planta</i>	0,0009	-0,0007
<i>Ganho do Controlador</i>	0,1882	-0,2379
<i>Tempo Integral</i>	0,0095	0,0075
<i>Tempo Derivativo</i>	0,0000	0,0000

Uma característica, que é necessária também controlar, ou pelo menos ter em consideração é a corrente. A resposta, em termos de corrente para o motor da direita com a aplicação das referências de velocidade utilizadas, encontra-se graficamente representada na figura 4.14.

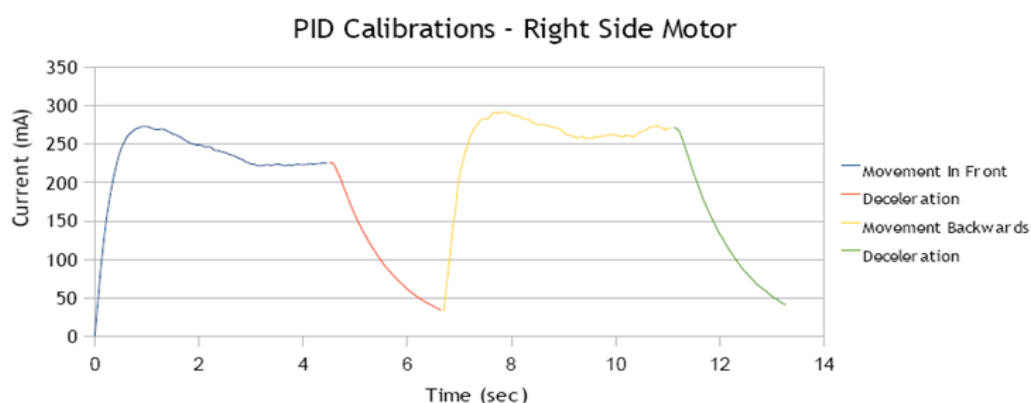


Figura 4.14: Resposta em Malha Aberta, Motor da Direita, em termos de Corrente

Podemos verificar que, em termos de corrente, o tempo de estabelecimento é muito superior ao que acontece em termos do gráfico da velocidade, sendo neste caso, de aproximadamente 3 segundos. É possível verificar a existência de “overshoot” no valor da corrente, que neste caso é de aproximadamente 22%.

Como o robot normalmente irá realizar deslocamentos no sentido positivo, iremos considerar os parâmetros para colocar no PID, como os valores apresentados para o sentido positivo.

Tabela 4.4: Parâmetros do PID utilizados para o Motor da Direita

<i>Ganho do Controlador</i>	0,1882
<i>Tempo Integral</i>	0,0095
<i>Tempo Derivativo</i>	0,0000

Com estes parâmetros, o motor da direita passou a ter as respostas em termos de velocidade e corrente, apresentadas respectivamente nas figuras 4.15 e 4.16.

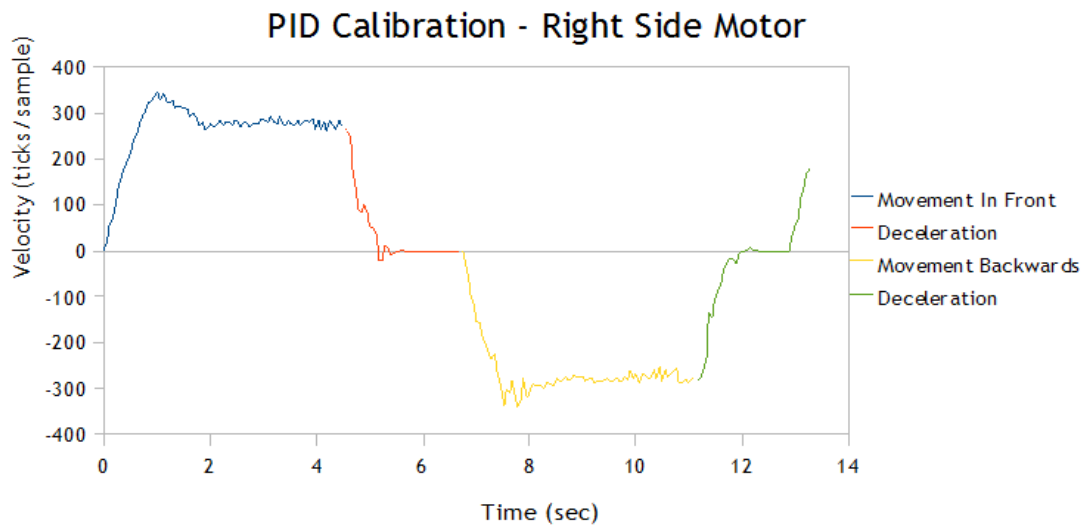


Figura 4.15: Resposta em Malha Fechada, Motor da Direita, em termos de Velocidade

Em termos de velocidade, verificamos que o erro em regime permanente, foi bastante reduzido mas, passou a existir um "overshoot" de velocidade aproximadamente de 24%.

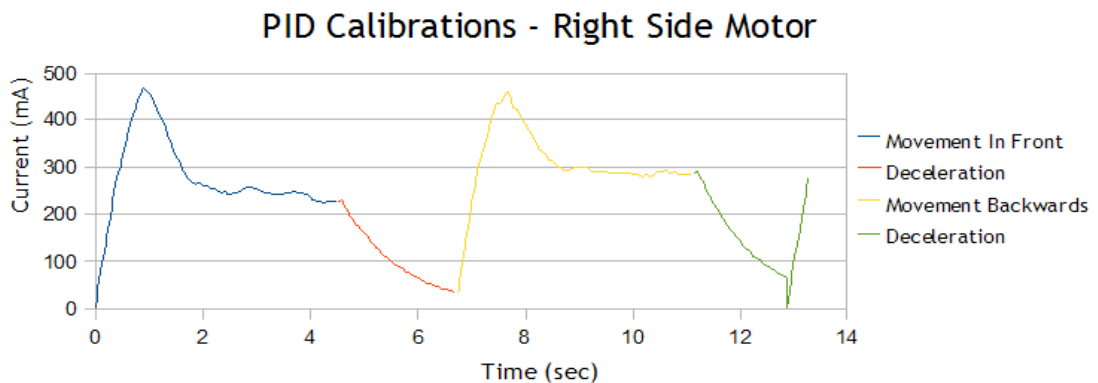


Figura 4.16: Resposta em Malha Fechada, Motor da Direita, em termos de Corrente

Em termos de corrente, verificamos que se me termos o valor médio da corrente este não é muito alterado, mas em termos de "overshoot" é aumentado consideravelmente passando agora para perto dos 85%. Mesmo assim, abaixo do limite de corrente aplicado.

Motor da Esquerda

Agora é a vez de analisar os dados referente ao motor da esquerda.

Inicialmente, este irá movimentar-se no sentido positivo considerado para a movimentação do robot, sendo então a velocidade medida negativa, passando depois num segundo momento a velocidade a ser positiva.

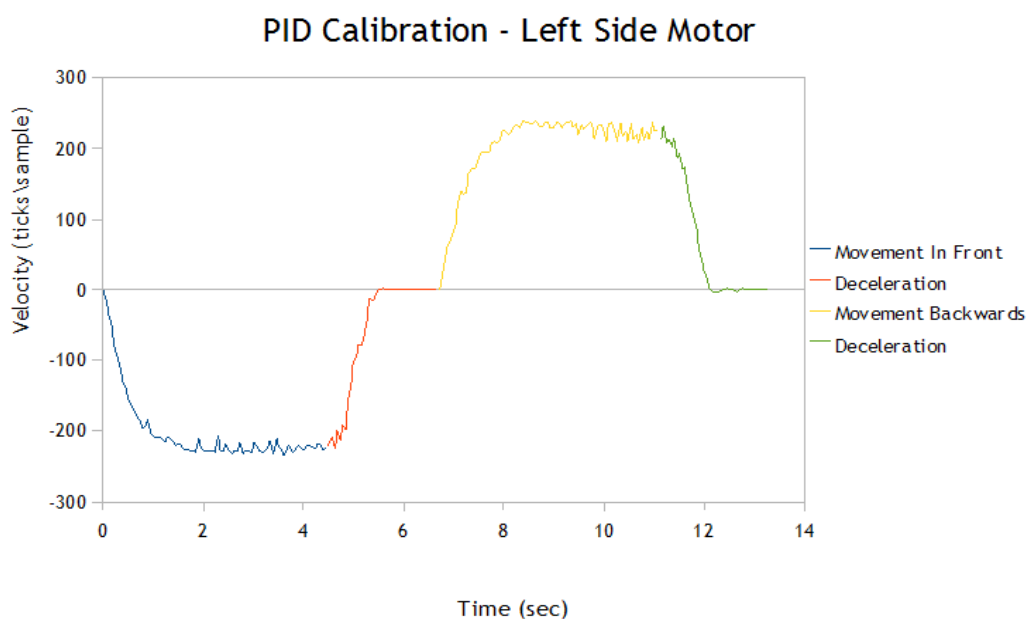


Figura 4.17: Resposta em Malha Aberta, Motor da Esquerda, em termos de Velocidade

É possível resumir os dados experimentais, obtidos para o motor da esquerda, e necessários para a calibração do controlador, sendo estes apresentados na Tabela 4.5.

Tabela 4.5: Resumo dos Principais Dados obtidos sobre a Dinâmica do Motor da Esquerda

<i>Características de Dinâmica para o Motor da Esquerda</i>	<i>Movimentação no Sentido Positivo</i>	<i>Movimentação no Sentido Negativo</i>
<i>Atraso (seg)</i>	0,1000	0,1000
<i>Valor Médio em Regime Permanente (ticks/amost)</i>	-219,8629	219,4519
<i>Tempo de Estabelecimento (seg)</i>	0,9629	0,8386

Tendo em consideração os dados apresentados na Tabela 4.5 e fórmulas de cálculo dos parâmetros de configuração do PID, segundo o Método Chien, presente na Tabela 4.1, podemos determinar os parâmetros a utilizar para o Motor da Esquerda, que são apresentados na Tabela 4.6.

Tabela 4.6: Resumo dos Dados obtidos para a Calibração do Controlador PID, segundo o Método de Chien, para o Motor da Direita.

<i>Método de Chien</i>	<i>Movimentação no Sentido Positivo</i>	<i>Movimentação no Sentido Negativo</i>
<i>Coeficiente de Chien</i>	2,0000	2,0000
<i>Ganho da Planta</i>	-0,0009	0,0008
<i>Ganho do Controlador</i>	-0,1891	0,2113
<i>Tempo Integral</i>	0,0096	0,0084
<i>Tempo Derivativo</i>	0,0000	0,0000

Uma característica, que é necessária também controlar, ou pelo menos ter em consideração é a corrente. A resposta, em termos de corrente para o motor da esquerda e com a aplicação das referências de velocidade utilizadas, encontra-se graficamente representada na figura 4.18.

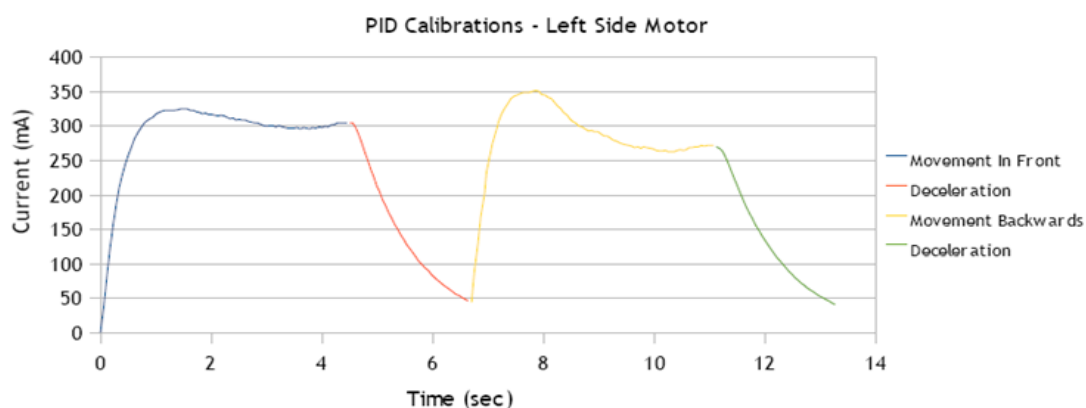


Figura 4.18: Resposta em Malha Aberta, Motor da Esquerda, em termos de Corrente

Podemos verificar que, em termos de corrente, o tempo de estabelecimento é muito superior ao que acontece em termos do gráfico da velocidade, sendo neste caso, de aproximadamente 3 segundos. É possível verificar a existência de “overshoot” no valor da corrente, sendo neste caso de 6% no caso da movimentação no sentido positivo e de 30% no sentido negativo.

Como o robot normalmente irá realizar deslocamentos no sentido positivo, iremos considerar os parâmetros para colocar no PID como os valores apresentados para o sentido positivo.

Tabela 4.7: Parâmetros do PID utilizados para o Motor da Esquerda

<i>Ganho do Controlador</i>	0,1891
<i>Tempo Integral</i>	0,0096
<i>Tempo Derivativo</i>	0,0000

Com estes parâmetros, o motor da esquerda passou a ter as respostas em termos de velocidade e corrente apresentadas respectivamente nas figuras

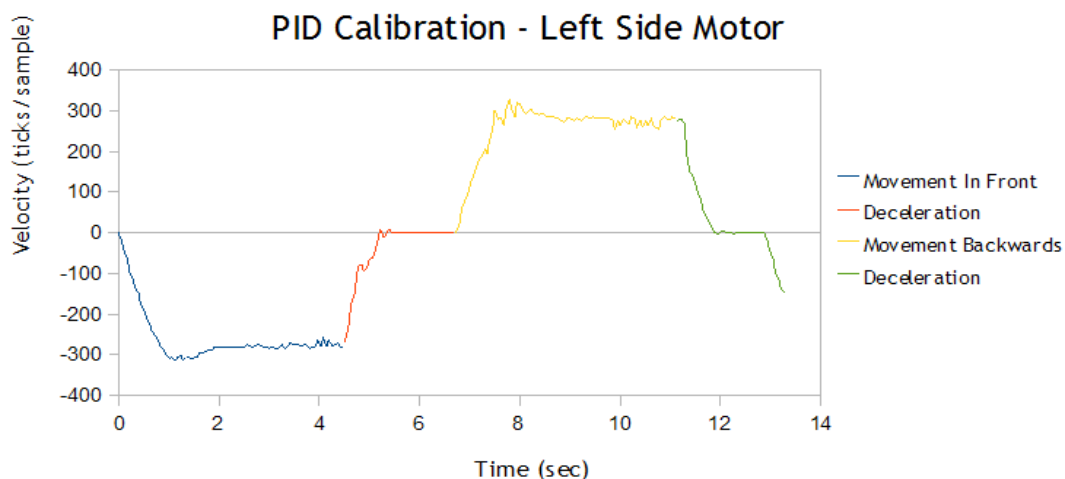


Figura 4.19: Resposta em Malha Fechada, Motor da Esquerda, em termos de Velocidade

Em termos de velocidade, verificamos que o erro em regime permanente foi bastante reduzido mas, passou a existir um "overshoot" de velocidade aproximadamente de 12%.

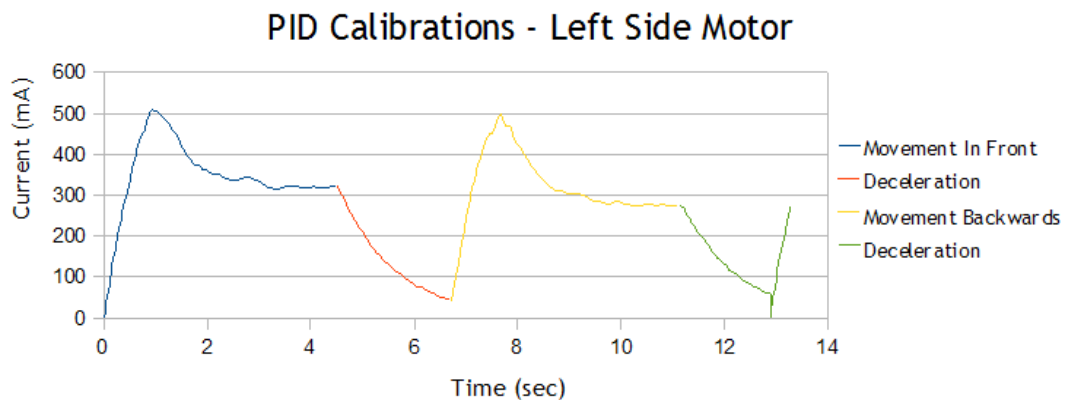


Figura 4.20: Resposta em Malha Fechada, Motor da Esquerda, em termos de Corrente

Em termos de corrente verificamos que, o valor médio da corrente, não é muito alterado, mas em termos de "overshoot" é aumentado consideravelmente, passando agora para perto dos 50%. Mesmo assim abaixo do limite de corrente aplicado.

4.2.3 Conclusões

Em suma, a calibração dos controladores PID vieram melhorar a performance dos motores e acima de tudo minimizar o erro entre a velocidade pretendida e as velocidades reais obtidas.

É necessário ter em consideração o valor da corrente máxima, especialmente porque os drivers de potência, apresentam uma limitação no valor de corrente máxima a fornecer aos drivers, sendo este valor de 1A.

Considerando que a velocidade normal de funcionamento do robot será a velocidade testada, foi possível comprovar que os limites máximos de corrente não são atingidos, mesmo após a calibração, que provocou uma maior intensidade na corrente pedida e, em consequência, um maior valor de “overshoot”.

4.3 Calibração da Odometria

4.3.1 Abordagem Implementada

A calibração da odometria é um procedimento no mínimo moroso e complexo. Para além do mais, não existe nenhum guia de como o fazer correctamente, nem equipamento adequado para tal nas instalações da faculdade.

A odometria, como referido anteriormente na secção de sensorização acerca dos encoders, baseia-se sobretudo em duas constantes físicas do robot, que são a distância entre rodas e o perímetro das mesmas.

O procedimento mais exacto para a calibração da odometria, seria executá-la com o recurso à estruturas que possibilitassem a sua realização com a menor interferência de factores externos e, com isto, maximizar a repetição de resultados. Para tal, o conveniente seria a implementação de uma estrutura de duas vias paralelas, como por exemplo (caminho de ferro), no qual o robot se deslocaria, obtendo apenas assim a componente tangencial da velocidade no robot e provocando que a velocidade angular fosse nula, devido a não haver rotação do robot em relação ao seu ponto central. Tal facto, pode ser constatado facilmente pelas equações 4.1 e 4.2 que definem, respectivamente, as velocidades tangencial e a angular do robot, através da velocidade tangencial de cada roda. O segundo passo seria efectuado através de uma estrutura circular com um raio adaptável, de forma a ser facilmente ajustado à distância entre as rodas do robot. Sendo assim, seria possível realizar a rotação do mesmo sobre o seu ponto central, calibrando o factor utilizado no cálculo da velocidade angular.

Dado não existir este tipo de equipamento/estrutura, recorreu-se a outro método de calibração, embora criativo, mas fiável e que minimizasse a possibilidade de erros. Para tal, utilizou-se uma intercepção de paredes rectas como base de referência de um sistema de coordenadas ortogonais. Delimitou-se na parede associada ao eixo das abcissas (XX's) dois pontos distanciados de 1 metro.

Utilizou-se duas almofadas de teflon colocadas no pára-choques do robot de forma a que este deslizasse facilmente usando a mesma parede como guia, como ilustrado na figura 4.21. Foi posicionado o centro do robot no primeiro ponto e deslocou-se o mesmo manualmente, até que o centro do robot se encontrasse no segundo ponto. Desta forma, foi adquirido o número total de ticks fornecidos por cada encoder associado à respectiva roda, e com a distância percorrida obteve-se a relação entre o perímetro de cada roda e o número de ticks. Esta relação indica-nos a distância equivalente a 1 tick. O motivo de existir duas constantes, cada uma destas associada a uma roda, prende-se ao facto das dimensões das rodas, polias serem ligeiramente diferentes assim como o posicionamento dos encoders.

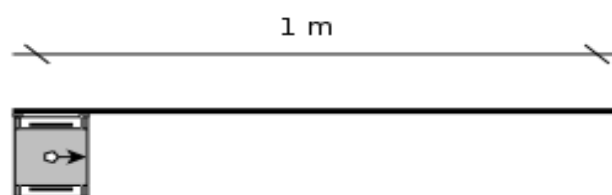


Figura 4.21: Esquema utilizado para a Calibração da odometria e Obtenção do Perímetro das Rodas

Resta-nos então a calibração do segundo factor, a distância entre rodas.

A calibração deste factor, após a calibração do perímetro das rodas, é realizada posicionando o robot a uma distância fixa da parede e com o recurso a um apontador laser posicionado no ponto central do robot, em que a sua projecção incide na origem de um quadro alvo fixado na parede, como se pode ver pela ilustração da figura 4.22.

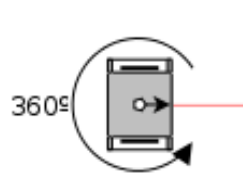


Figura 4.22: Esquema utilizado para a Calibração da odometria e Obtenção da Distância entre Rodas

O robot foi rodado 360° em torno do seu eixo com o auxilio de uma calha circular e posicionado com a ajuda do laser na origem do quadro de alvos, medindo-se assim os ticks fornecidos pelos encoders em semelhança à calibração dos perímetros das rodas. Assumindo também que o ponto central do robot, não se deslocou, a trajectória formada na rotação corresponde a uma circunferência com o diâmetro igual à da distância entre rodas. Então, usando a distância percorrida por estas (ambas distâncias idênticas), obtemos a distância entre rodas, com o recurso à formula de cálculo do perímetro de uma circunferência.

$$P_{circulo} = 2\pi \cdot r \Leftrightarrow P_{circulo} = \pi \cdot d \quad (4.8)$$

4.3.2 Dados Experimentais

Após as calibrações acima descritas, por mais tentativas que se realizem é praticamente impossível conseguir que a odometria seja perfeita, devido a erros relativos à resolução do encoder, tempos de amostragem, variação do ponto de contacto com o piso, variação da distância entre rodas, devido à presença de uma placa metálica de suporte aos motores, que provoca um leve amortecimento e ao deslizamento das rodas. Para tal, à semelhança da calibração dos perímetros das rodas e da distância entre estas, utilizou-se a intercepção das paredes como referência num eixo de coordenadas ortogonal.

O robot foi posicionado a uma distância X_i e Y_i entre as respectivas paredes consideradas referenciais e o seu ângulo θ_i definido como nulo, como demonstrado na figura 4.23. Então, através do programa de controlo do robot, foi ordenado que esta percorresse uma distância de 1m a uma velocidade baixa e, no final foram obtidos os respectivos valores, tendo inclusive repito este processo várias vezes.

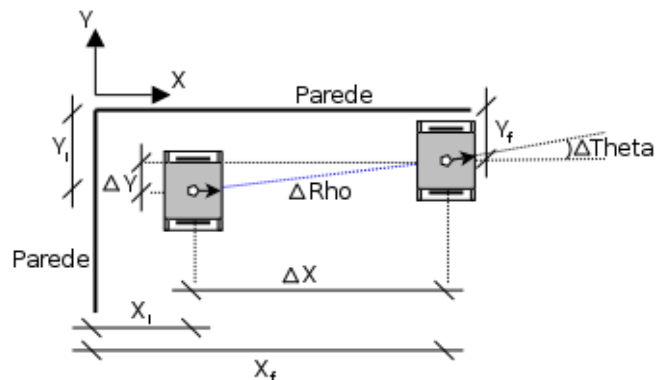


Figura 4.23: Modelo para a Determinação do erro de Translação e do Deslizamento

Os dados obtidos por esta experiência encontram-se apresentados na tabela 5.1 em anexo e permitem a formulação dos modelos associados ao movimento de translação e do deslizamento do robot.

O deslizamento do robot é caracterizado por um desvio no ângulo do

robot, provocado pela assimetria dos pesos dos diversos componentes que o compõem, bem como pelas diferentes respostas por parte dos motores que originam um comportamento diferente na translação do mesmo.

Para o modelo da odometria no que respeita às componentes de translação e deslizamento, a distribuição que mais se adequa é a distribuição normal pela caracterização dos erros como se poderá observar pela figura (Gráfico dos Dados), onde se denota uma tendência da propagação dos erros dentro de um determinado intervalo. Dado que nesta experiência foi definido antecipadamente um deslocamento de 1m, os valores obtidos de erro já estão normalizados nas unidades do sistema internacional.

Tabela 4.8: Modelo de odometria - Erro de Translação

Modelo de odometria - Erro de Translação			
[m/m]	X	Y	Rho
Média	0,0212	-0,0139	0,0212
Desvio Padrão	0,0040	0,0035	0,0040

Tabela 4.9: Modelo de odometria - Erro de Deslizamento

Modelo de odometria - Erro de Deslizamento	
[rad/m]	Theta
Média	-0,0013
Desvio Padrão	0,0027

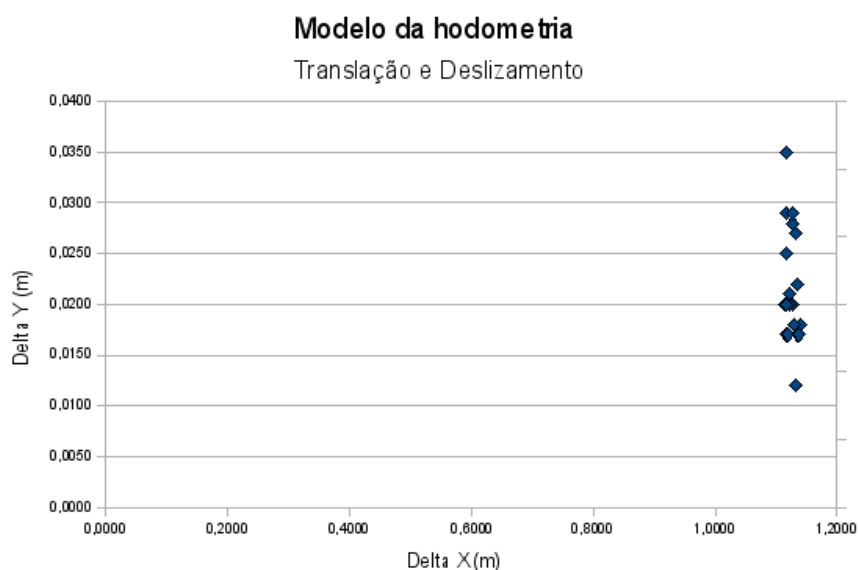


Figura 4.24: Representação Gráfica dos dados Experimentais para o Modelo de Translação e Deslizamento

Para a modelização da rotação, posicionou-se o robot à frente de uma parede e com recurso a um ponteiro laser, acertou-se a sua orientação de forma a ficar coincidente com a origem de um quadro de alvos. Através do programa de controlo ordenou-se ao robot para perfazer uma volta completa em torno do seu ponto central, como ilustrado pela figura 4.25.

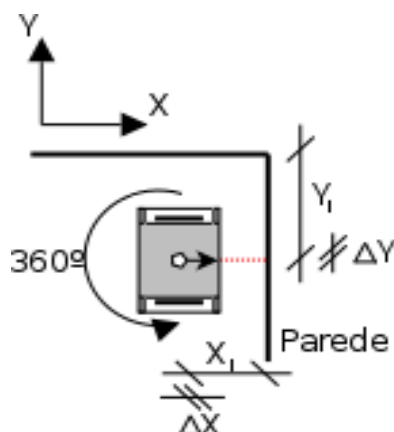


Figura 4.25: Esquema utilizado para a obtenção do Modelo do Erro na Rotação

Dado que o robot não conseguiu rodar em torno do seu eixo, tendo existido sempre um deslocamento deste, optou-se por compensar este deslocamento, recalculando o theta final com base nestes deslocamentos, aproximando-se bastante do theta obtido pelo robot.

Os dados retirados estão compilados na tabela 5.4 em anexo.

Tal como no modelo da translação e do deslizamento, o modelo da rotação pode ser descrito como uma distribuição normal facilmente visualizada no gráfico da figura 4.26.

Tabela 4.10: Modelo de odometria - Erro de Rotação

Modelo de odometria - Erro de Rotação	
[rad/2π]	Theta
Média	-0,0086
Desvio Padrão	0,0068

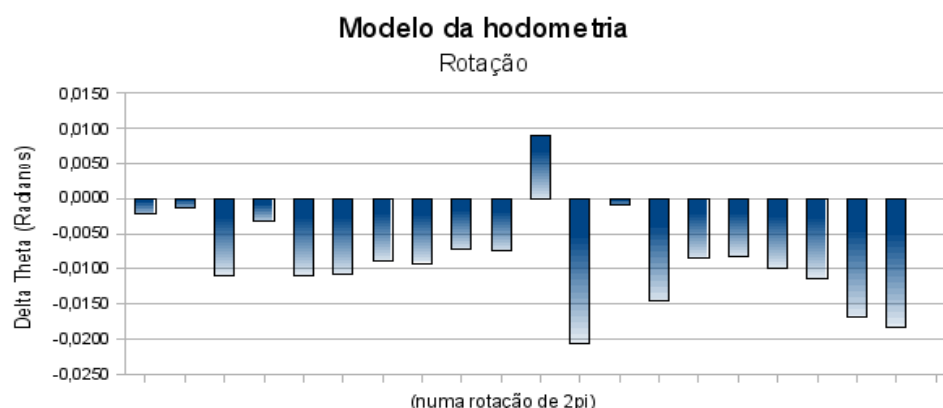


Figura 4.26: Representação Gráfica dos Dados Experimentais para o Modelo de Rotação

4.3.3 Conclusões

A odometria como fonte de informação possui uma importância que não pode ser ignorada. A sua calibração não é realizada de uma forma simples, nem poderá ser exacta, possuindo inúmeros erros que não são possíveis de ser eliminados, quanto muito, no melhor esforço, poderão ser minimizados.

Os erros de odometria possuem a particularidade de serem acumulativos, por outras palavras, os erros são aproximadamente lineares conforme a distância percorrida. Isto significa que, recorrendo a um modelo matemático, poderemos prever dentro de uma certa gama de valores, os erros que a odometria poderá assumir. Desta forma, podemos obter um certo grau de confiança sobre a fiabilidade dos valores desta, reutilizando-os numa fase posterior em algoritmos de fusão de informação, de forma a validar ou rejeitar, a informação transmitida por esta.

4.4 Modelo do Erro dos Sonares

A determinação do modelo dos Sonares foi realizada por um colega de trabalho, no mesmo projecto, porém numa outra vertente de fusão de informação, pelo que transcrevo aqui um excerto da sua tese.[13]

4.4.1 Abordagem Implementada

Para a obtenção do modelo do sonar conduziu-se a duas experiências, num ambiente controlado, num espaço amplo e estático, sem que houvesse interferência de objectos nas mediações do sonar com a excepção do objecto alvo de testes. Foi posicionado o sonar a cerca de 40cm do piso, numa posição paralela à superfície.

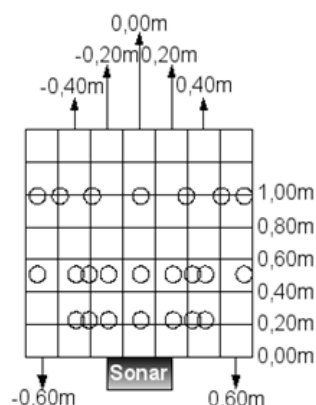


Figura 4.27: Grelha da disposição do Objecto Cilíndrico de Testes em relação ao Sonar

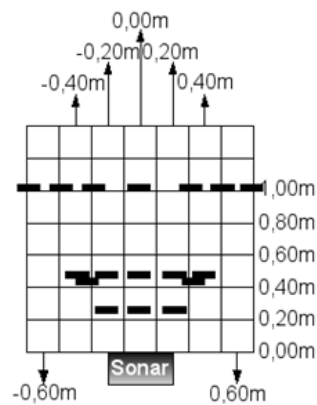


Figura 4.28: Grelha da disposição do objecto plano de testes em relação ao Sonar

Numa primeira fase foram registadas cerca de 380 medidas a um objecto cilíndrico de acrílico com um diâmetro de 10cm por uma altura aproximada de 50cm posicionado em diferentes distâncias do sonar, figura 4.21. Os valores obtidos foram registados estando na página web da dissertação. Na segunda fase, foram registadas cerca de 340 medidas a um objecto com uma superfície lisa de madeira contraplacada com uma altura de 50cm e uma largura de 10cm, sendo este colocado também a diferentes distâncias, conforme a figura 4.22 demonstra.

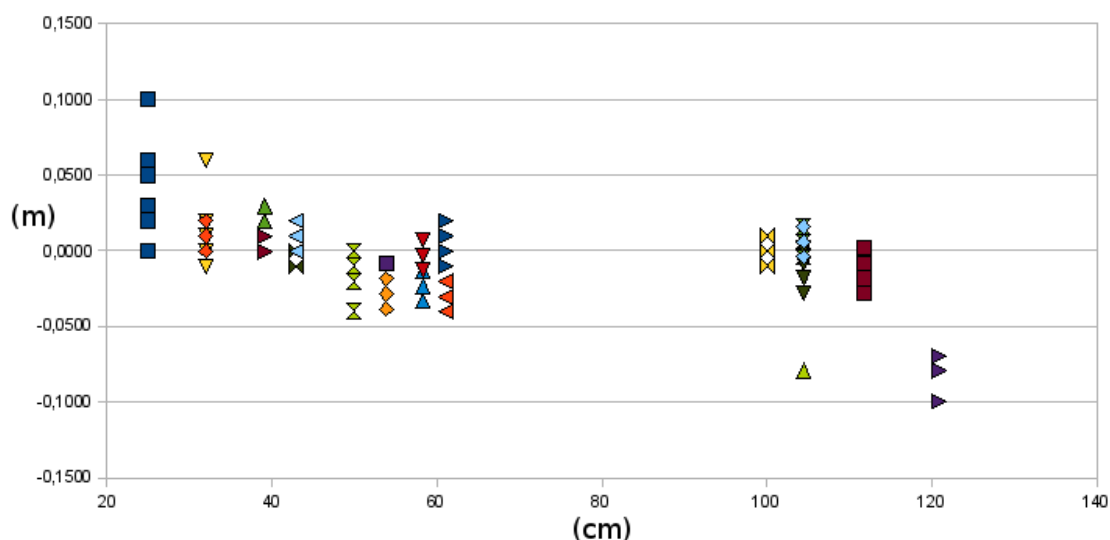


Figura 4.29: Representação gráfica dos erros absolutos em relação à distância no teste com um cilindro de acrílico

Nas figuras 4.23 e 4.24, encontram-se gráficos, que representam os erros absolutos em relação às distâncias nos quais foram efectuadas as medições. Na experiência realizada com o cilindro, observa-se que os piores resultados resultam, quando o objecto está na proximidade do sonar, embora existam resultados fracos quando este se situa a cerca de 1,20m, o qual se deve à

reflexão do eco proveniente da reflexão no piso.

Contudo, é seguro afirmar, que para uma gama de valores entre 0cm a 120cm o erro situa-se nos 10cm. Por outras palavras, existe um erro percentual de 8,3%, que torna claro que este sensor não é o ideal para medidas de distâncias a paredes, quando comparados com outros sensores igualmente de baixo custo.

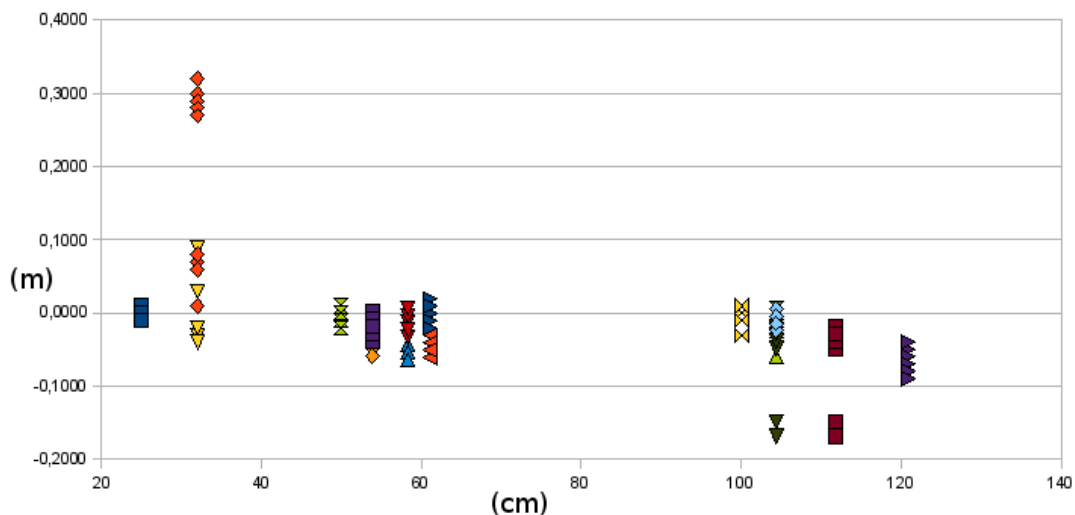


Figura 4.30: Representação gráfica dos erros absolutos em relação à distância no teste com uma placa de contra-placado

Quando analisados os erros correspondentes ao teste com a placa de contra-placado, a análise demonstra a mesma evolução, com a excepção de algumas medições, em que a placa reflecte os ecos recebidos para uma direcção diferente da original, originando assim o efeito de deflexão. Não entrando em linha de conta com as medidas sujeitas a este tipo de efeito, o erro é semelhante à experiência anterior, havendo um erro acentuado quando o objecto se encontra na próxima do sensor mas, estando dentro de um erro de 10 cm na gama de valores entre 0 a 120cm.

Tabela 4.11: Modelo do Erro dos Sonares

Modelo dos Sonares	
[m / m]	Rho
Média	0,0088
Desvio Padrão	0,0579

4.4.2 Conclusões

Este tipo de sonar comunica por interface I2C e possibilita uma certa margem de manobra a nível de configuração do alcance e ganho. Contudo constatou-se que mesmo provocando um ganho nulo através da sua configuração, este tipo de sonar ainda possuía uma sensibilidade razoável, tornando-se indesejável no disparo dos seis sonares em conjunto originando interferência na captação de ecos entre eles, implicando assim medidas incorrectas. Por outro lado o seu grande ângulo de dispersão do eco, torna-o ideal para a detecção de obstáculos. No entanto, esta mesma característica deixa-o à mercê de irregularidades no solo a uma certa distância, dando origem a medidas incorrectas.

Uma outra desvantagem do uso de sonares, é o tempo de aquisição de uma medida, para um alcance de 6m o tempo de aquisição é de 65ms, para ir de encontro com a abordagem já referida de inúmeros sensores e dado à interferência entre eles é necessário, criar uma relação de compromisso, reduzindo o seu alcance, bem como desenvolver métodos que permitam o disparo numa sequência alternada, de forma a que não haja interferências entre os ecos de cada sonar, na tentativa de reduzir o tempo de aquisição.

Face a estas desvantagens todas, é que se utilizará este tipo de sensor? Como referido anteriormente, este é um sensor ideal para a detecção de obstáculos, podendo ser usado pelo path planing na decisão de alteração de rotas para situações anómalas.

4.5 Arquitectura de Software

Para a realização de um controlo e decisão no Robot e Limpeza, foi definida uma arquitectura de software robusta, que utiliza as potencialidades de uma linguagem “event-driven” - orientada a eventos -, tal como o Lazarus. A arquitectura definida proporcionou a melhor interligação possível com todos os elementos que compõem o projecto.

Tendo por base o programa de controlo e decisão desenvolvida para o Robot de Limpeza, é possível definir que os principais eventos, encontram-se representados na figura 4.31.

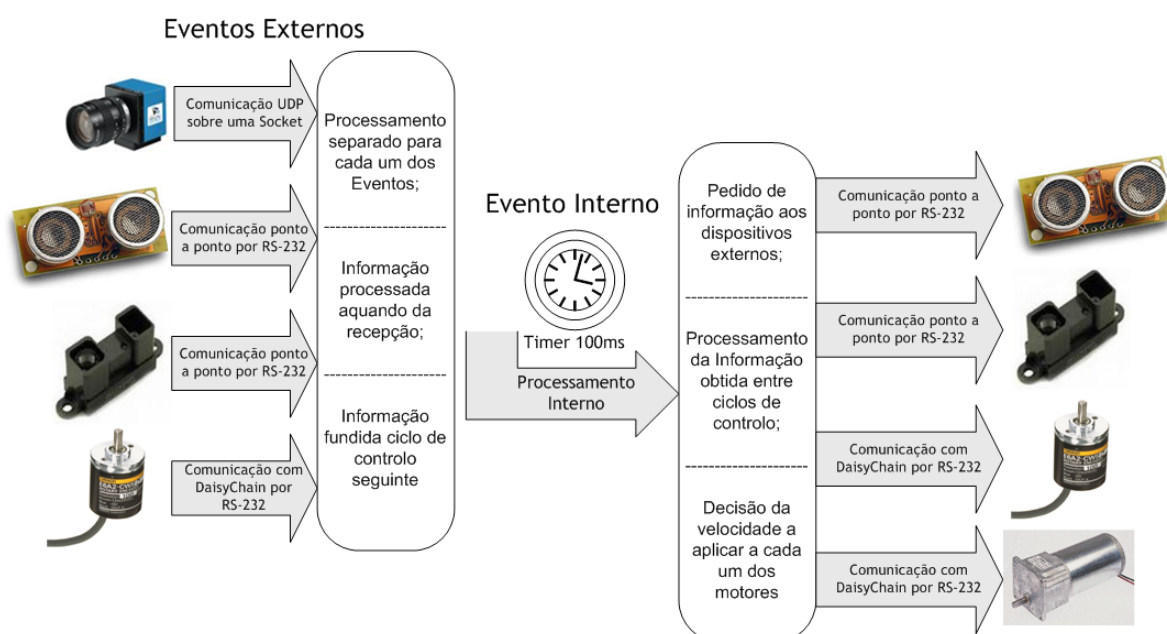


Figura 4.31: Principais Eventos presentes no Programa de Controlo e Decisão do Robot de Limpeza

Cada evento externo é processado de forma independente e assíncrona, ou seja, após a recepção do evento, e logo que o programa termine todas as acções que se encontrava a fazer, é que é propriamente processado. O instante de início do seu processamento, assim como o tempo de processamento, são calculados e guardados.

O evento interno, ou seja, o timer que controla o tempo entre ciclos de controlo, é lançado periodicamente de 100 em 100ms. É durante o ciclo de controlo que todas as decisões e processamentos importantes são realizados. Os principais blocos de acções presentes no ciclo de controlo, encontram-se apresentados na figura 4.32.

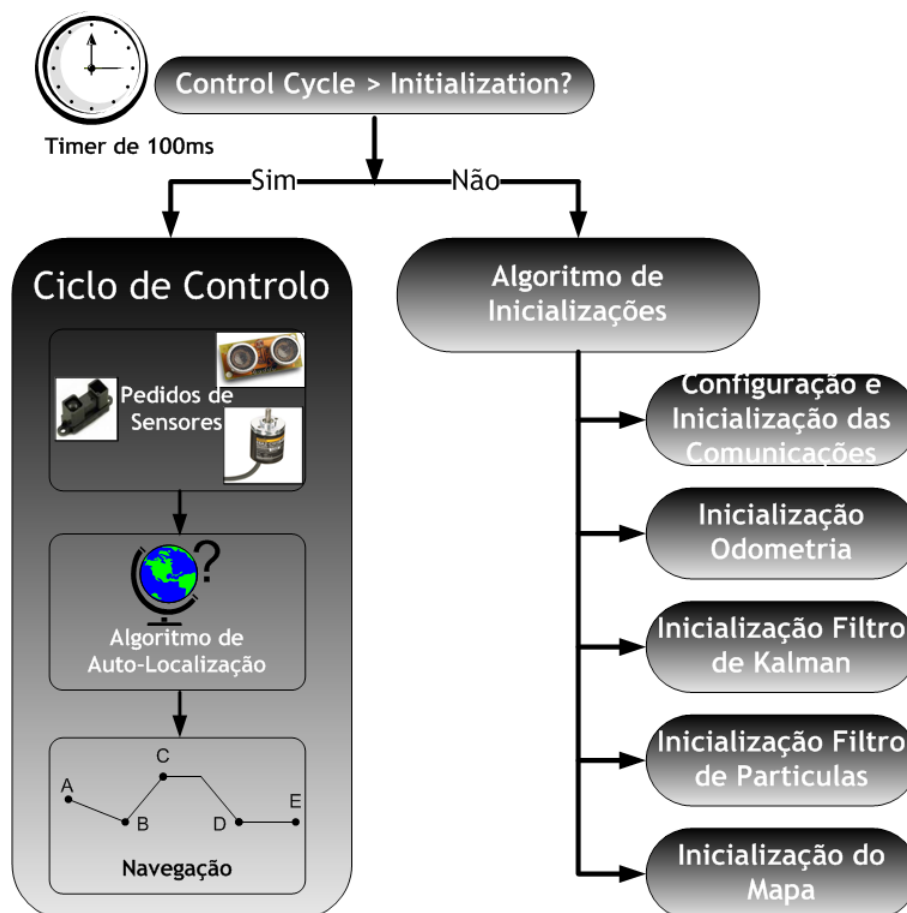


Figura 4.32: Esquema em Blocos das acções a tomar no Ciclo de Controlo

Nas primeiras entradas no ciclo de controlo, é necessário realizar algumas inicializações, tanto ao nível das comunicações com os dispositivos externos, com reconfiguração de alguns parâmetros, como dos métodos de auto-localização presentes no Robot de Limpeza (Filtro de Kalman e Filtro de Partículas), ou ainda, a leitura do ficheiro de mapa da área que o Robot de Limpeza irá limpar, não só para o conhecimento dos códigos de barras da área, paredes, mas também, para se poder utilizar como ferramenta de debug ou de controlo visual da posição actual do Robot de Limpeza.

Passado todas estas inicializações, o Robot da Limpeza está pronto para se começar a movimentar, realizando sempre os blocos apresentados no lado esquerdo da figura 4.32.

Primeiro de tudo, são realizados os pedidos de medidas dos diversos dispositivos externos, de modo a apresentar, no próximo ciclo de controlo, informações novas sobre o estado actual do Robot de Limpeza. Feito isto, agora é necessário estimar a posição actual e qual a acção a realizar, acções estas em que são responsáveis os dois outros blocos principais dentro do ciclo de controlo.

Em relação ao bloco de auto-localização, as suas funções encontram-se esquematizadas na figura 4.33.

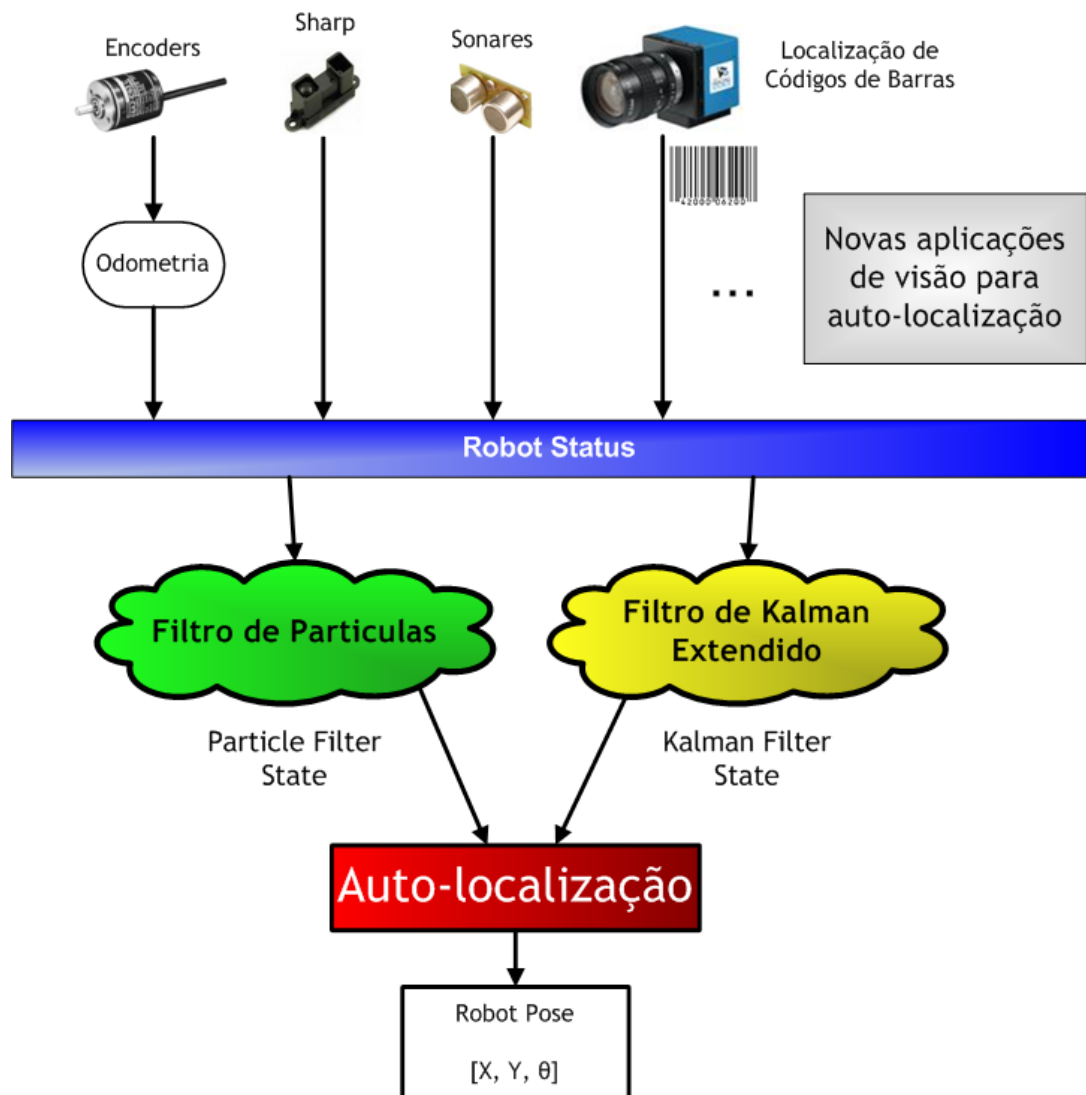


Figura 4.33: Mecanismos de Auto-Localização no Robot de Limpeza

A informação sobre o estado dos diversos sensores encontra-se acessível na estrutura *Robot Status*. Os métodos de auto-localização utilizam essa informação, validam e integram-na nos seus processamentos, afim de chegar a uma estimativa para a posição actual do Robot de Limpeza.

Os processos realizados pelo bloco do Filtro de Kalman Extendido são apresentados de uma forma mais exaustiva neste documento, no capítulo 4.6.1.

Em relação ao Filtro de Partículas, a sua aplicação no Robot de Limpeza pode ser investigada recorrendo a dissertação do meu colega de projecto responsável por esse método. [13]

Por fim, e voltando ao esquema da figura 4.32 temos a fase de Navegação. Aqui, foi implementado um sistema de path planning baseado em "checkpoints" de posições globais, que o Robot de Limpeza terá que passar durante a sua missão.

A relação entre a posição estimada do Robot de Limpeza e a posição do "checkpoint" é então processada e recorrendo às funções de movimentação são definidas as velocidades tangencial e angular a aplicar ao Robot de Limpeza, de modo a que este consiga chegar ao seu objectivo.

Todo o funcionamento do programa de controlo e decisão, encontra-se sustentado por uma definição global de uma estrutura que representa uma verdadeira compilação de toda a informação presente no Robot de Limpeza.

```
type TStatus=record
  TimeStamp: Dword;
  case StatusData: integer of
    0: (OdometryData: TOdometry);
    1: (SonarData: TSonar);
    2: (BarCodeData: TBarCode);
    3: (SharpData: TSharp);
    4: (KalmanFilterData: TKalmanFilter);
    5: (ParticlesFilterData: TParticlesFilter);
    6: (RobotInfo: TRobotInfo);
end;
```

A escolha da utilização de uma estrutura tipo "union" para o Robot Status, foi decidida devido ao assincronismo dos eventos externos e que num mesmos ciclo de controlo não é certo, que todas as sub-estruturas tivessem que ser preenchidas.

Devido a este facto, e como para um mesmo ciclo de controlo, mais que uma sub-estrutura é escrita, ou seja, uma informação diferente é avaliada. Este facto levou a duas novas decisões. Primeiro a introdução da variável TimeStamp, que define o instante em que a estrutura foi escrita e ainda, uma forma em array circular para a variável Robot Status, o que promove o controlo do tamanho máximo e possíveis usos desnecessários ou descontrolados da memória.

A definição da estrutura do Robot Status, proporciona, pela sua forma e divisão, a criação de um "Log", ou seja, um histórico sobre o estado de todas as variáveis do sistema. Esta função de "Log", encontra-se no Robot de Limpeza implementada de duas formas: uma de modo "online", devido a forma em array circular da estrutura e do uso de variáveis globais, que guardam a posição do array o ultimo estado processado do sistema, podendo-se assim analisar o estado anterior de qualquer uma das variáveis, de qualquer uma das sub-estruturas, e também para um modo "offline" já que no fim de cada ciclo de controlo é escrito num ficheiro e de forma estruturada, toda a informação continua nas últimas posições de cada umas das sub-estruturas.

Este ficheiro de "Log" criado, pode ser então, depois da realização de uma missão, ser analisado instante a instante. Para esse fim, foi criada uma outra aplicação, chamada de "Log Replay", criada com a função de leitura e análise dos ficheiros criados pelo programa de controlo e decisão do Robot de Limpeza.

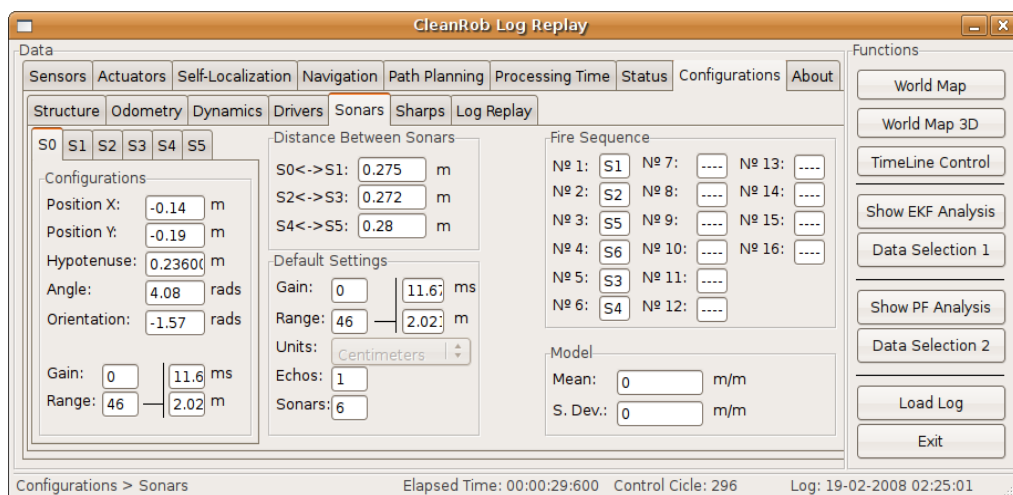


Figura 4.34: Interface Gráfica da Aplicação Log Replay do Projecto do Robot de Limpeza

Da figura 4.34 é possível perceber, que toda a informação obtida da missão realizada pode ser então analisada por esta aplicação.

É possível também traçar gráficos 2 e 3D, e controlar a movimentação temporal do Robot de Limpeza, proporcionando uma análise instante a instante se assim for pretendido, ou de uma forma mais rápida, ou mesmo só de um instante definido. Tudo isto é controlado pela interface apresentada na figura 4.35.



Figura 4.35: Interface de Time Line Control da Aplicação Log Replay

Em suma, todas as aplicações realizadas no projecto do Robot de Limpeza, partilham as mesmas estruturas de controlo, o que facilita o desenvolvimento de novas aplicações de auxílio e uma base de trabalho comum.

4.6 Auto-localização por Filtro de Kalman

4.6.1 Abordagem Implementada

Para a realização da auto-localização no robot de limpeza foi utilizada uma abordagem baseada no Filtro de Kalman Extendido.

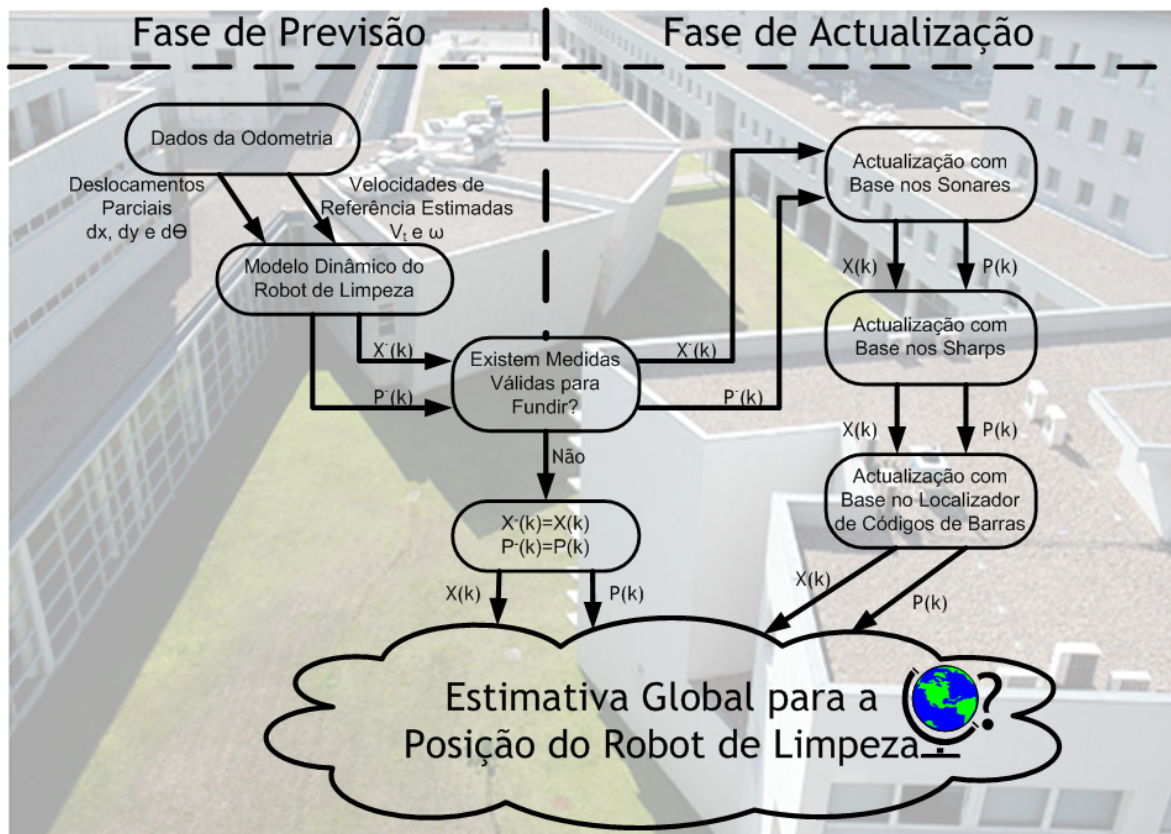


Figura 4.36: Acções a Realizar durante a Estimação via Filtro de Kalman Extendido

Em termos de implementação real, esquematizada na figura 4.36, foi utilizados os dados proveniente da odometria para a realização da previsão do estado, sendo os outros elementos de sensorização, utilizados na fase de actualização.

A actualização é realizada sempre que existem medidas validas. A validação das medidas é determinado por um filtro de verosimilhança, que testa os limites das medidas, avaliando cada uma das medidas e determinando a sua admissão ou não, para a realização da fusão de informação com a previsão realizada.

Como foi dito acima, a actualização é só realizada quando existem medidas verosímeis, o que implica, que nem sempre se realizem todos os tipos de actualizações. Quando não existem, a estimativa $X(k)$ e a confiança

$P(k)$ não são actualizadas passando para o bloco de actualização seguinte.

Em termos de sequência de operações, esta é sempre realizada das medidas com pior modelo para o erro, para a medidas com melhor modelo, tendo em consideração também a frequência esperada para a obtenção de medidas válidas. Assim sendo, primeiro realiza-se a actualização das medidas dos sonares, seguidas pelas medidas dos sharps terminando com a informação da localização do código de barras.

Mas antes de se realizar as implementações práticas no robot, foram realizadas simulações em Matlab, afim de comprovar e estudar melhor os argumentos teóricos do Filtro de Kalman.

4.6.2 Simulações

Filtro de Kalman Discreto

Para completar o estudo e por forma a compreender melhor o Filtro de Kalman Discreto, realizamos simulações em Matlab de um sistema linear teste.

Este sistema era composto por um robot que se movimentava sobre carris, adquirindo movimento por acção de uma força. Deste sistema tínhamos acesso às medidas da força aplicada, entrada do sistema, às medidas de posição no eixo “xx” e velocidade instantânea, correspondentes aos estados do robot e, consequentemente, aos estados do sistema.

De modo a simplificar os cálculos e a análise do sistema, consideramos que o robot tem a massa de um quilograma, o que faz com que em termos de magnitude a aceleração e força aplicada, apresentem o mesmo valor.

$$\begin{cases} p_{k+1} = p_k + \Delta t \times v_k \\ v_{k+1} = v_k + \Delta t \times u_k \end{cases} \quad (4.9)$$

Resultando numa representação em espaço de estados discreto:

$$\begin{cases} \begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \cdot u_k \\ \begin{bmatrix} y_p \\ y_v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} \end{cases} \quad (4.10)$$

Como referido na parte introdutória do Filtro de Kalman, para iniciar o algoritmo é necessário apresentar uma estimativa inicial tanto para o estado x_0 como para a covariância P_0 , sendo as utilizadas para as simulações:

$$x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} ; P_0 = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \quad (4.11)$$

Os valores escolhidos para preencher a matriz P_0 não são despropositados. É normal que a estimativa inicial não seja completamente precisa. No caso específico das simulações, onde é comum saber-se o estado inicial do sistema de uma forma precisa, pretende-se, mais do que tudo, verificar as características do método, velocidade e modo de convergência, de forma a se conseguir inferir sobre a robustez do método. Para se conseguir esse efeito é normal colocar-se, inicialmente, valores altos na matriz P_0 como os apresentados em 4.11.

Em termos de definição de convergência do método, esta é normalmente considerada como a aproximação da estimativa com a posição “real” do robot no carril.

Estas medidas, no nosso caso, são ambas conhecidas, já que a posição “real” é determinada inicialmente pela simulação, conjugando a informação da entrada, que é predefinida e do modelo dinâmico do sistema 4.12. Esta informação poderá ser depois disponibilizada como medida na parte da actualização do Filtro de Kalman.

$$\begin{cases} y_{p\ k+1} = y_{p\ k} + \Delta t \cdot y_{v\ k} \\ y_{v\ k+1} = y_{v\ k} + \Delta t \cdot u_k \end{cases} \quad (4.12)$$

O algoritmo utilizado para o Filtro de Kalman é o apresentado na Tabela 3.2.

Após as considerações apresentadas acima, estamos prontos a apresentar as simulações e as conclusões adquiridas.

Para as simulações foram criadas dois cenários de teste:

- **Primeiro Cenário** - as medidas nem sempre estão disponíveis, mas quando existem são de boa qualidade;
- **Segundo Cenário** - a meio da simulação (aproximadamente aos 5 segundos), o robot é raptado, sendo colocado no instante seguinte numa posição diferente. Algum tempo depois, o robot é novamente raptado (aproximadamente aos 7,5 segundos). Neste cenário as medidas são sempre de boa qualidade.

Em termos da entrada do sistema durante a simulação, ou em termos representativo, a força que é aplicada ao sistema ou consequente aceleração, é a seguinte:

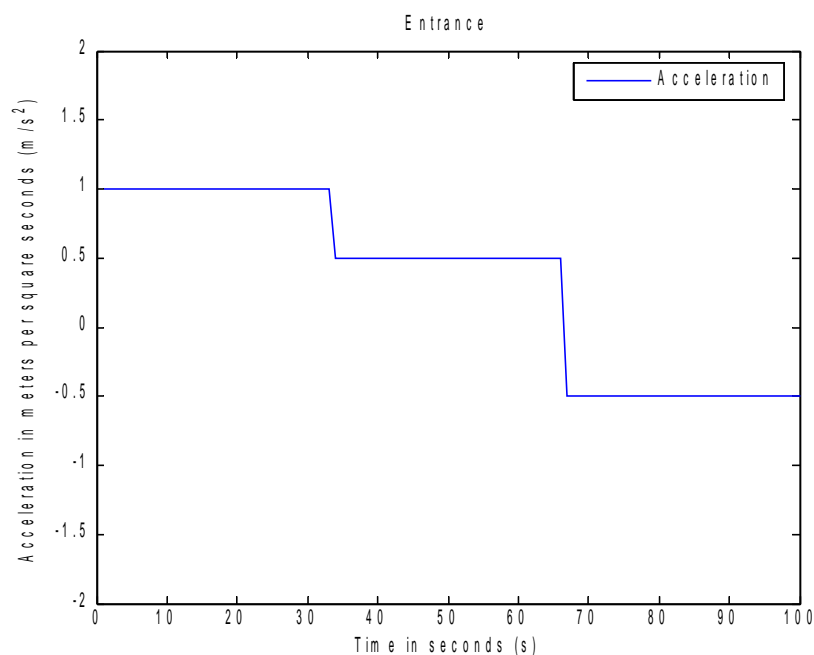


Figura 4.37: Entrada (força => aceleração) ao longo do tempo

Primeiro Cenário em Simulação

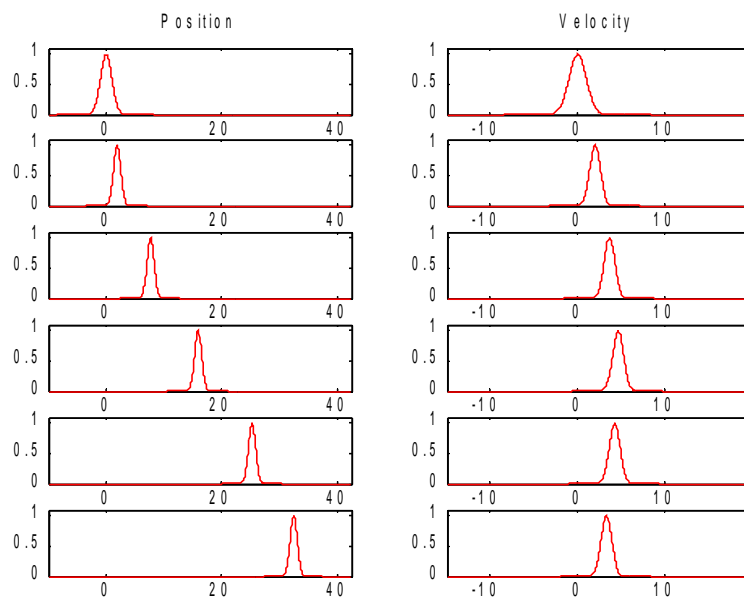


Figura 4.38: Gaussianas representativas de Posição e Velocidade - Cenário 1

Neste cenário, testamos a robustez do Filtro de Kalman em termos de resposta a ausência de medidas válidas, já que uma medida em cada série de cinco é considerada não válida.

O desenvolvimento temporal, é apresentado de forma vertical nas figuras 4.38 e 4.41.

O método converge, se a diferença entre a posição estimada e “real” tender para zero ou estabilizar num valor baixo, em termos de dados da simulação este fenómeno é verificado pelo emagrecimento progressivo das curvas ao longo do tempo.

Na figura 4.38 é possível verificar a convergência do método, expressado pelo emagrecimento das curvas.

É também importante referir, que se alterarmos os valores das matrizes Q e R a forma das curvas é também alterada. Quanto menor forem os valores das matrizes Q e R, mais delgadas as curvas se tornaram.

Analisemos agora o erro de posição e velocidade apresentados nesta simulação:

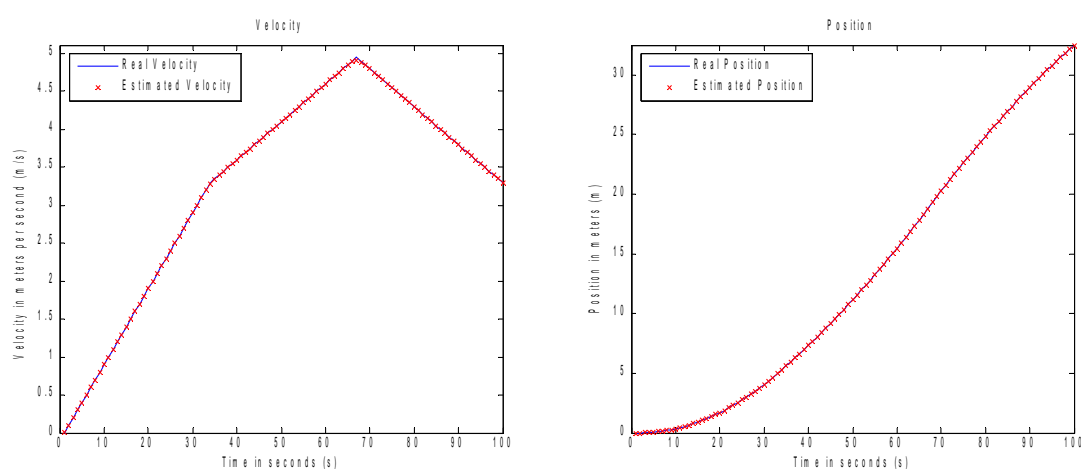


Figura 4.39: Comparação entre a posição e velocidade estimados e “reais” - Cenário 1

Com os gráficos apresentados na figura 4.39 podemos observar a qualidade da estimativa conseguida com o uso do Filtro de Kalman.

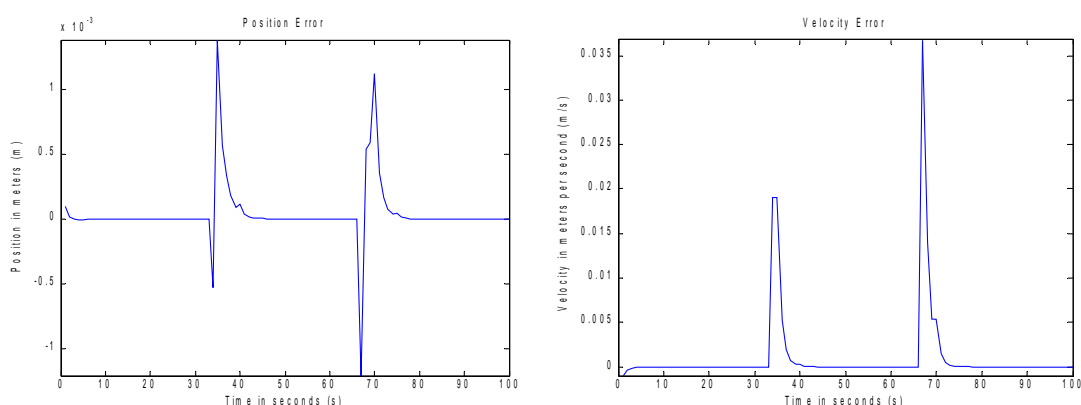


Figura 4.40: Erros entre a estimativa e medida “real” para posição e velocidade - Cenário 1

Neste caso, o erro presente encontra-se mesmo muito próximo de zero, sendo somente superior nos instantes de transição de valor na entrada.

Segundo Cenário em Simulação

Neste segundo caso, iremos analisar um problema muito importante em termos da robótica, o caso do rapto do robot.

Este teste, poderá representar, erros de posição causados por embates, deslizamentos ou erros de odometria.

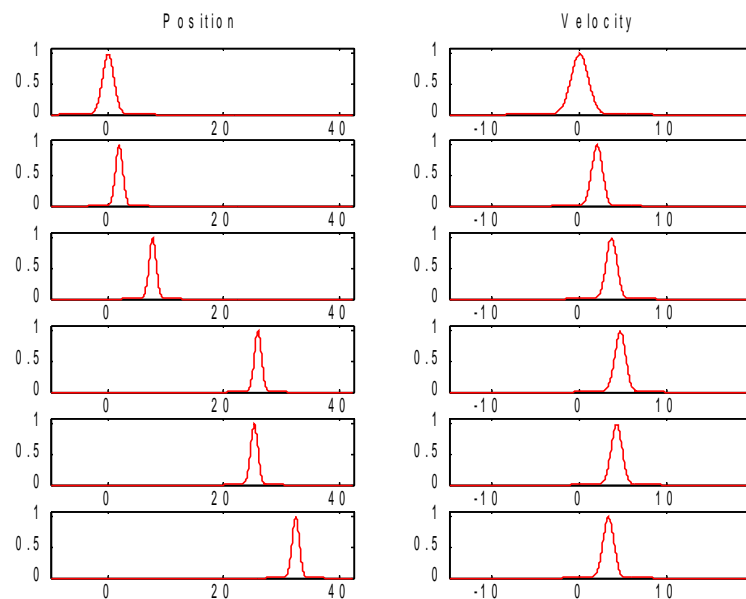


Figura 4.41: Gaussianas representativas de Posição e Velocidade - Cenário 2

Em termos conceptuais, aproximadamente a meio do caminho o robot é “teleportado” da sua posição actual para uma 10 metros mais à frente, algum tempo depois, o robot volta a ser raptado, mas agora o robot é recuado 10 metros.

As curvas conseguidas com a simulação, no que tem a ver com a forma, são semelhantes às do cenário anterior, em que as diferenças são causadas pelo rapto que vêm alterar a posição final do robot, assim como da velocidade, sendo neste ultimo ponto menos expressiva. Em termos da espessura das curvas, estas parecem-se quase iguais.

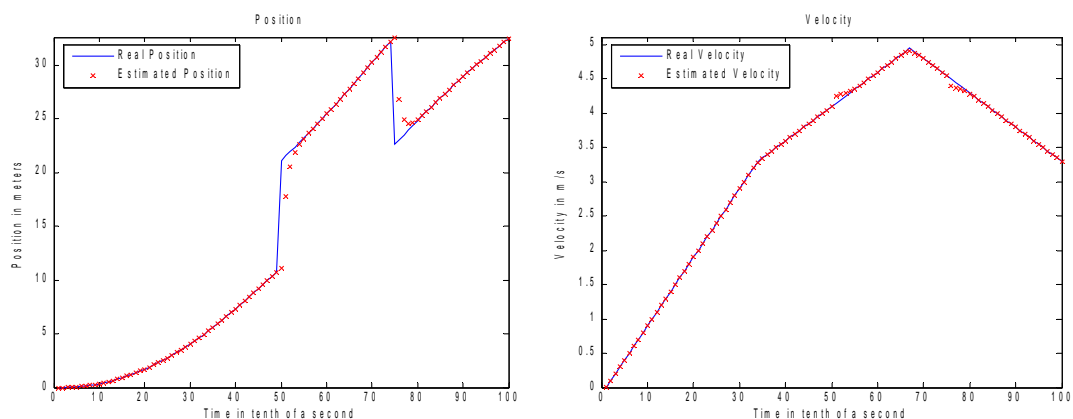
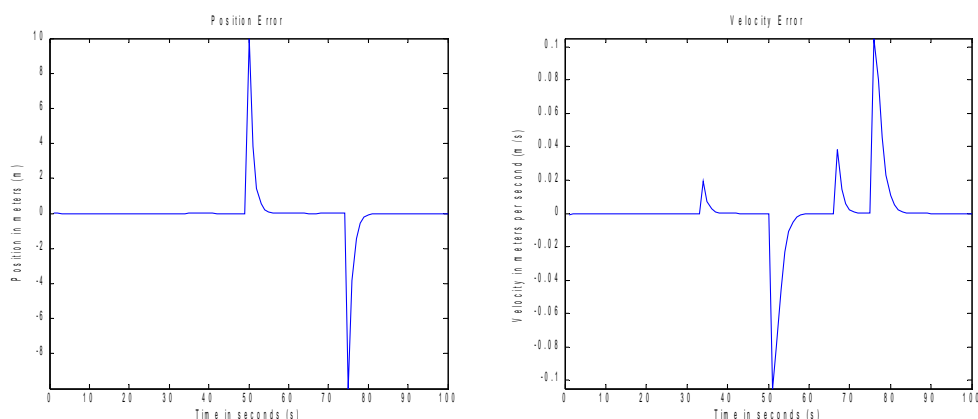


Figura 4.42: Comparação entre a posição e velocidade estimados e “reais” - Cenário 2

A posição estimada do robot não se altera de um forma instantânea, já que na perspectiva do Filtro de Kalman o movimento de “teleporte” é impossível. Sendo assim, a única resposta que o Filtro de Kalman poderá executar é em relação à velocidade, é modificando-a de forma a conseguir expressar a alteração de posição. Proporcionando assim, que o método volte a convergir para a posição “real”. A rapidez de convergência do método varia com a capacidade de modificação da velocidade do robot.



Figura

4.43: Erros entre a estimativa e medida “real” para posição e velocidade - Cenário 2

Nos instantes do rapto, erros em termos de posição e velocidade são consideráveis, mas são ambos recuperados nas iterações a seguir.

É sempre bom recordar, que o sistema é linear e que nos encontramos a utilizar o Filtro de Kalman, que nestes casos converge sempre, sendo mesmo a solução óptima se o ruído apresentar média nula.

As figuras 4.42 e 4.43, apresentam de uma forma visual e muito mais compreensível, as conclusões retiradas nos três últimos parágrafos, já que é notório o aumento da velocidade afim de compensar o erro na posição entre a posição “real” e a estimativa realizada.

Filtro de Kalman Extendido

Para o caso do Filtro de Kalman Discreto, completamos o nosso estudo com algumas simulações em Matlab de um sistema onde o Filtro de Kalman Extendido se aplica.

Neste caso, o sistema implementado era composto por um robot, de arquitectura cadeira de rodas, que se deslocava pelo segundo andar do Edifício I, da Faculdade de Engenharia da Universidade do Porto, através de velocidades predefinidas que são consideradas como as entradas do sistema.

Em relação à posição e covariância iniciais necessárias para o método iterar, realizam-se nos mesmo moldes, que no caso discreto, uma posição inicial correcta mas numa covariância elevada.

$$x_0 = \begin{bmatrix} PosXIni \\ PosYIni \\ PosTetaIni \end{bmatrix} ; P_0 = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} ; U = \begin{bmatrix} V \\ W \end{bmatrix} \quad (4.13)$$

Novamente, na lógica de funcionamento da simulação utilizada para o caso discreto, as posições reais do robot são calculadas, sendo esta informação fornecida, na fase de actualização do Filtro de Kalman, quando pretendemos uma medida válida.

$$\begin{cases} x_{k+1} = x_k + V_{k+1} \cdot \cos(\Theta_k) \cdot \Delta t \\ y_{k+1} = y_k + V_{k+1} \cdot \sin(\Theta_k) \cdot \Delta t \\ \Theta_{k+1} = \Theta_k + W_{k+1} \cdot \Delta t \end{cases} \quad (4.14)$$

O algoritmo utilizado para Filtro de Kalman Extendido é o apresentado na tabela 3.3.

Tratando-se de um sistema não linear, é necessário determinar a matriz A responsável pelo modelo do sistema em cada iteração, tendo esta como representação base:

$$A_k = \begin{bmatrix} 0 & 0 & -v(t_k) \sin(\theta(t_k)) \\ 0 & 0 & v(t_k) \cos(\theta(t_k)) \\ 0 & 0 & 0 \end{bmatrix} \quad (4.15)$$

$$A = \exp(A(k)(t(k) - t(k-1)))$$

Para as simulações foram criadas dois cenários de teste:

- **Primeiro Cenário** - as medidas nem sempre estão disponíveis, mas quando existem são de boa qualidade;
- **Segundo Cenário** - a meio da simulação (aproximadamente aos 25 segundos), o robot é raptado, sendo colocado no instante seguinte numa posição diferente. Algum tempo depois, o robot é novamente raptado

(aproximadamente aos 37,5 segundos). Neste cenário as medidas são sempre de boa qualidade.

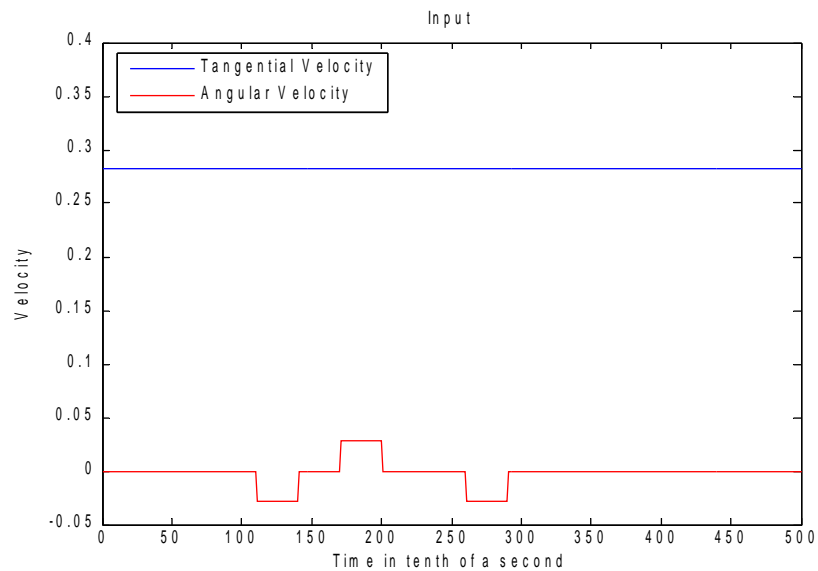


Figura 4.44: Entrada do sistema

Em termos da entrada do sistema teremos as seguintes velocidades em cada instante de amostragem, em que a azul está representada a velocidade tangencial e a vermelho a velocidade angular.

Primeiro Cenário em Simulação

Neste cenário, voltamos a testar a robustez do Filtro de Kalman em termos de resposta à ausência de medidas válidas, já que uma medida, em cada série de cinco, é considerada não válida, o que inviabiliza a fase de actualização no Filtro de Kalman.

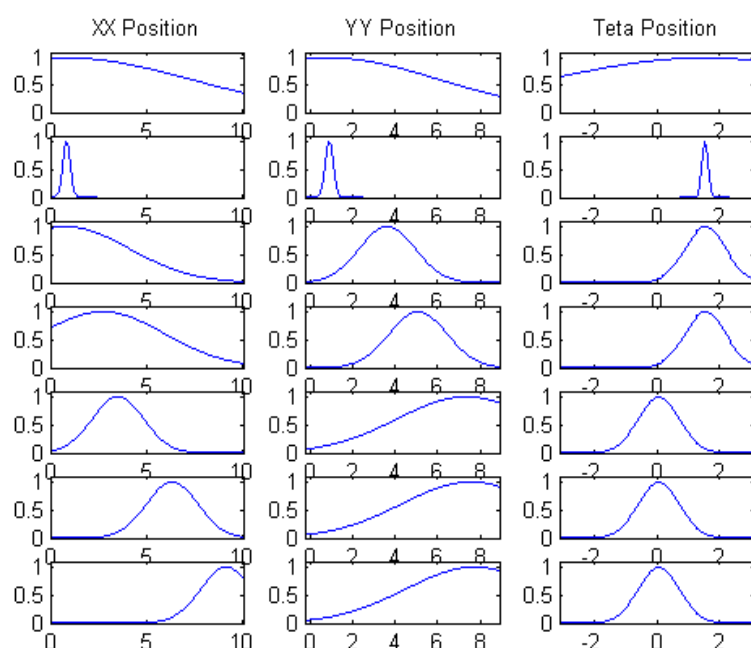


Figura 4.45: Gaussianas representativas da posição do robot - cenário 1

Antes de começar com a análise dos resultados é necessário explicar que as figuras 4.45 e 4.53 apresentam um desenvolvimento temporal de cima para baixo, em que as duas primeiras representam os dois primeiros décimos de segundo, a terceira o instante dez segundos e os restantes distanciados dez segundos do anterior.

Iniciando a análise podemos verificar que, o Filtro de Kalman, do primeiro para o segundo momento consegue, logo, melhorar imenso a confiança que o método apresenta, na estimação realizada, demonstrando a rápida taxa de convergência do método.

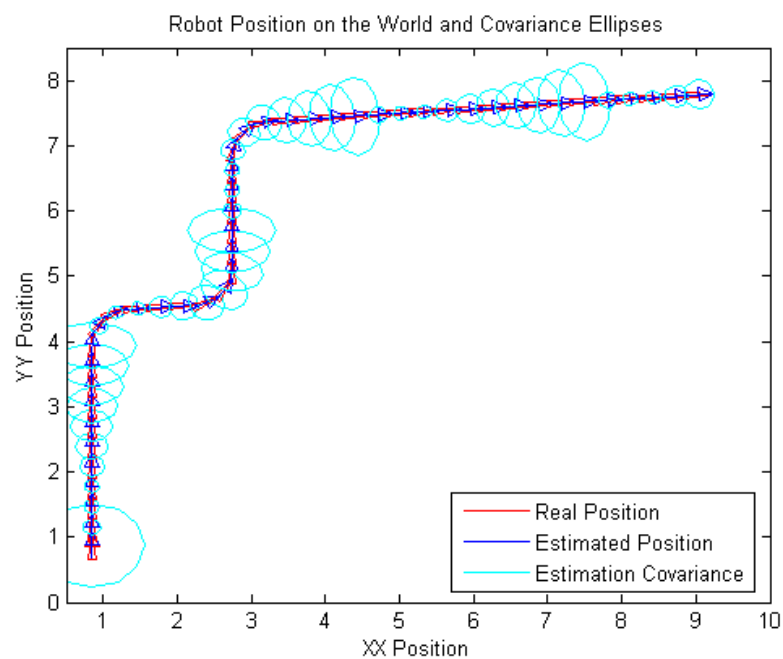


Figura 4.46: Covariâncias da Posição Global do Robot - Cenário 1

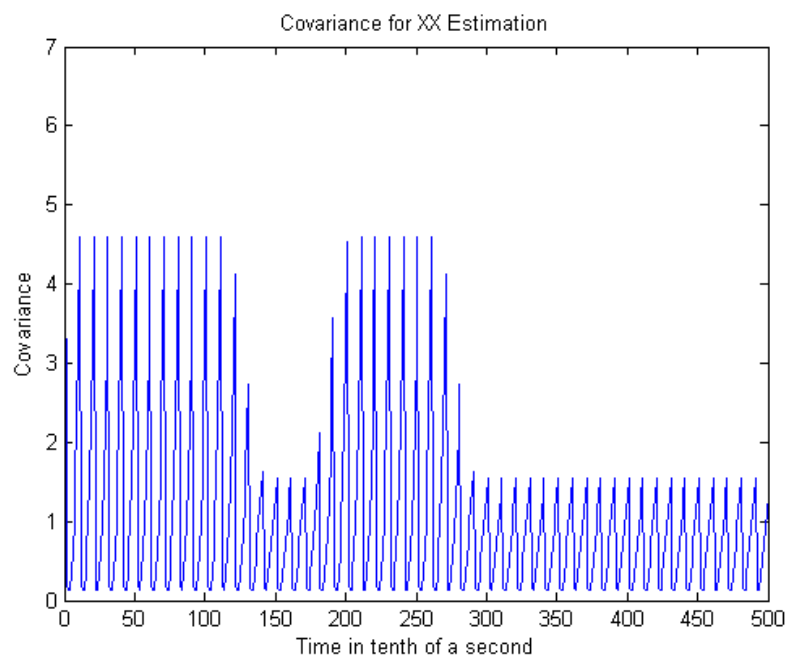


Figura 4.47: Covariâncias da Posição do Robot XX - Cenário 1

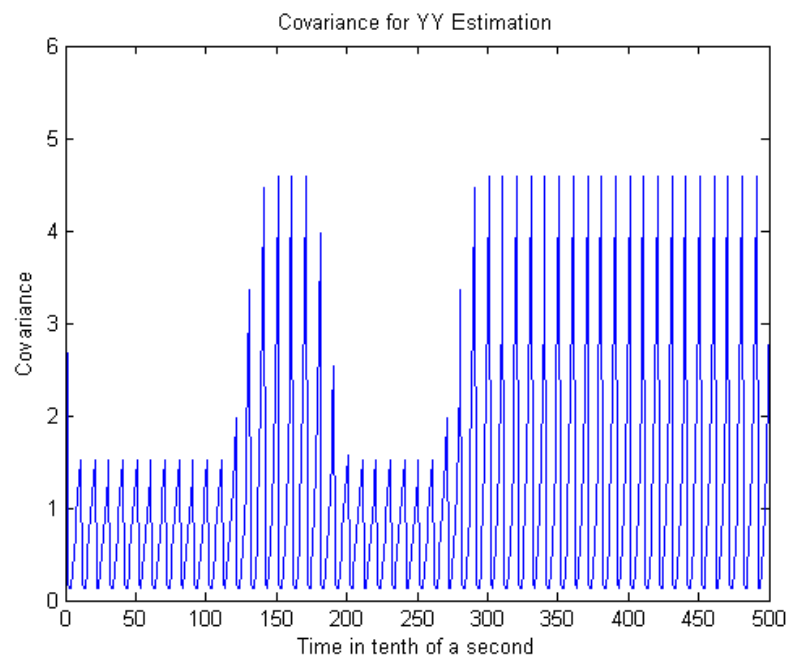


Figura 4.48: Covariâncias da Posição do Robot YY - Cenário 1

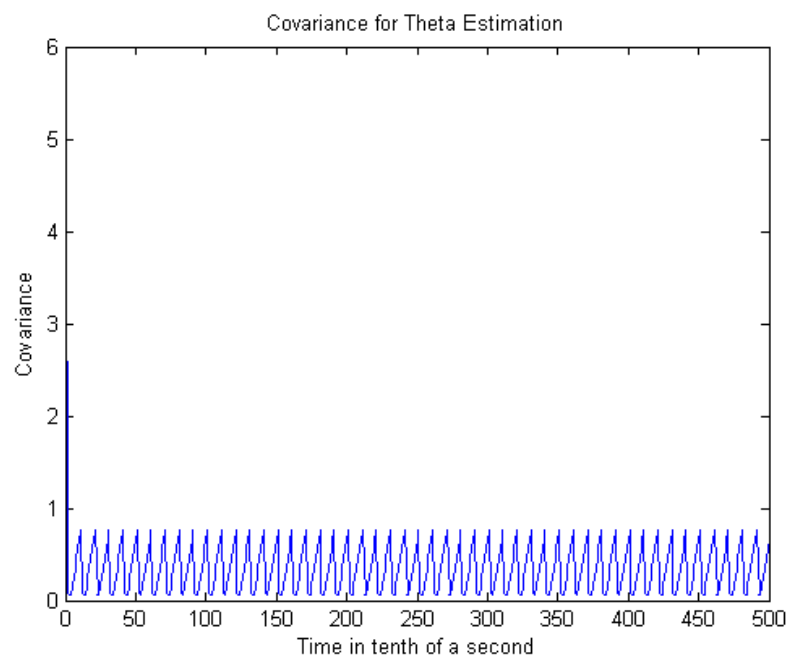


Figura 4.49: Covariâncias da Posição do Robot em Theta - Cenário 1

A presença de medidas não válidas afecta a confiança do método, não é extremamente visível na figura 4.49, mas verificados os valores para os elementos da matriz P apresentados no Matlab, é possível verificar que na sua estimação, o que provoca que a covariância cresça, mas com a elevada taxa de convergência do método, no décimo de segundo seguinte, o algoritmo volta a realizar estimações boas e a acreditar-se mais na sua estimação.

De seguida, apresentamos algumas imagens da movimentação do robot no mundo, mapa do segundo piso do edifício I da Faculdade de Engenharia da Universidade do Porto:

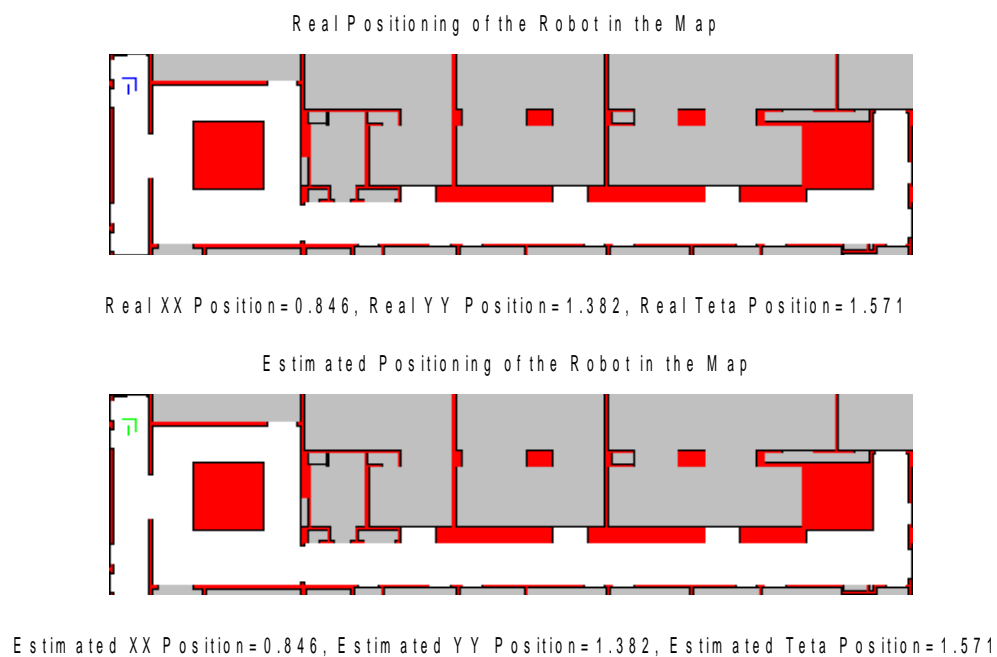


Figura 4.50: Representação simulada no mapa - cenário 1

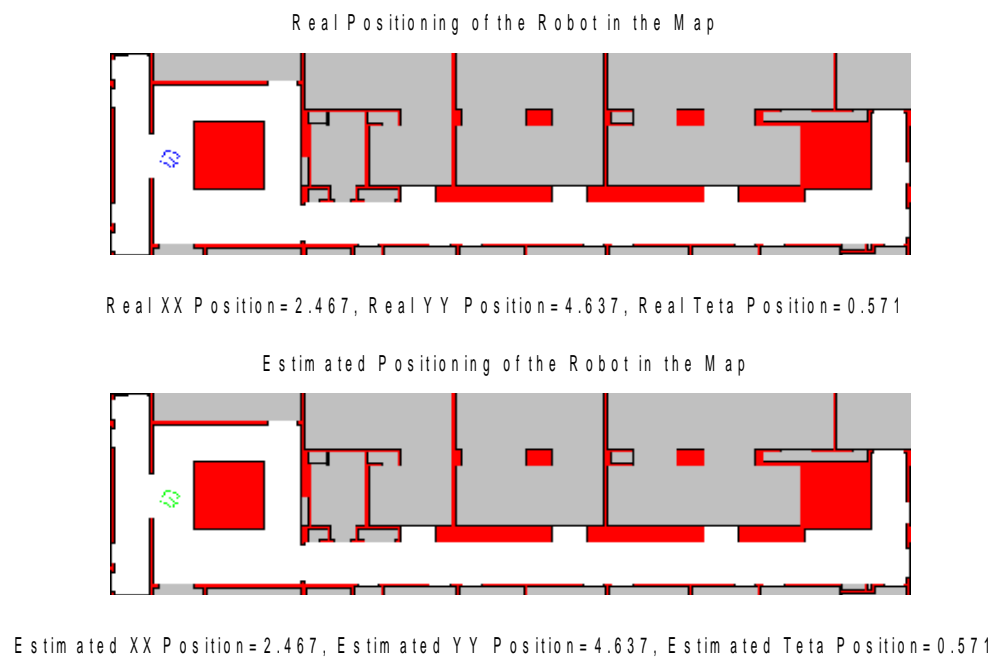


Figura 4.51: Representação simulada no mapa - cenário 1

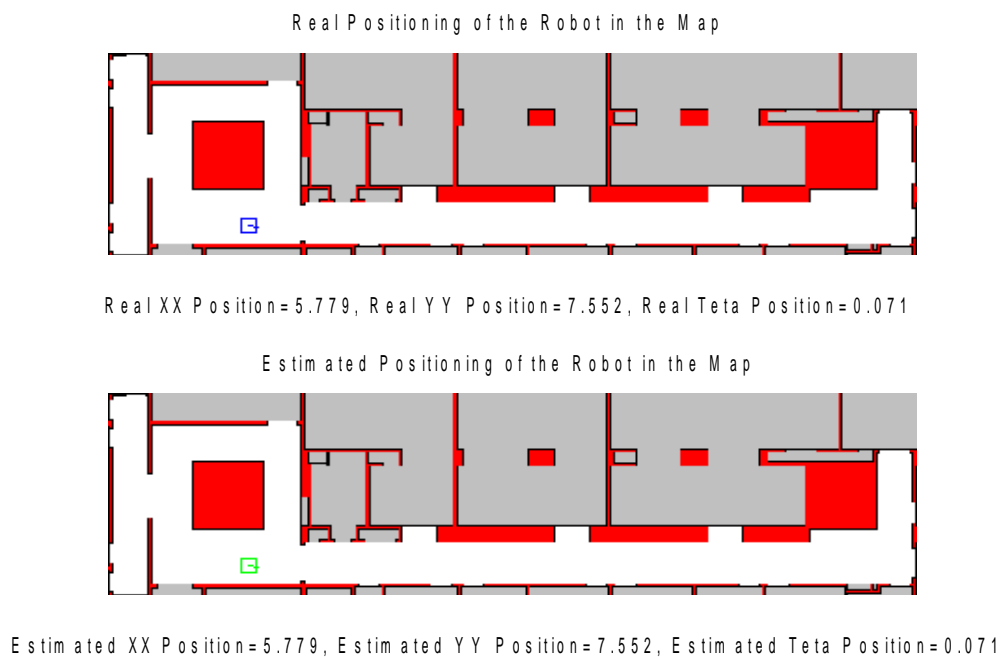


Figura 4.52: Representação simulada no mapa - cenário 1

Das figuras 4.50, 4.51 e 4.52 podemos nos aperceber da qualidade da estimação realizada pelo Filtro de Kalman, estando sempre, se não muito próximo, exactamente coincidente com os valores reais da posição do robot.

Segundo Cenário em Simulação

No segundo cenário simulado, abordamos um problema muito comum na robótica, o caso do rapto do robot.

Este pode ser representativo de erros de odometria, deslizamentos de rodas ou mesmo choques que o robot possa sofrer.

No caso simulado o robot sofre dois raptos, um os 25 segundos e outro os 37,5 segundos.

Analisemos agora a confiança na estimativa no mesmo conjunto de instantes utilizados na simulação anterior e representados na figura 4.45.

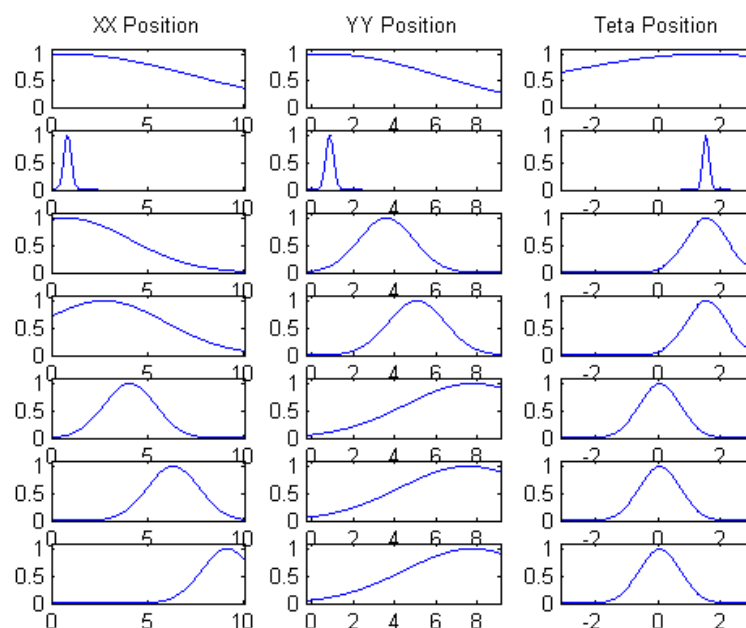


Figura 4.53: Gaussianas representativas da posição no cenário 2

A presença de medidas sempre válidas, provoca que os valores de covariâncias sejam menores, proporcionando curvas mais elegantes.

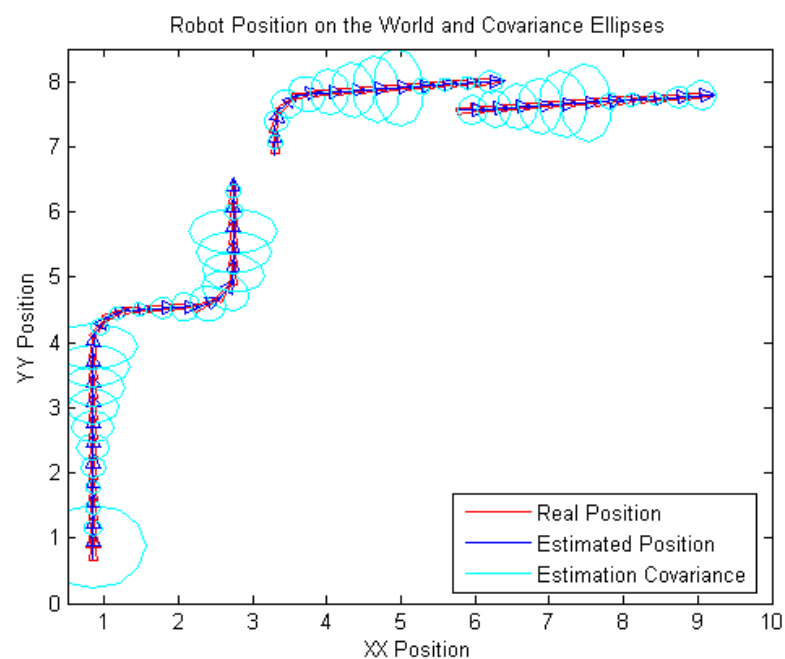


Figura 4.54: Covariâncias da Posição Global do Robot - Cenário 2

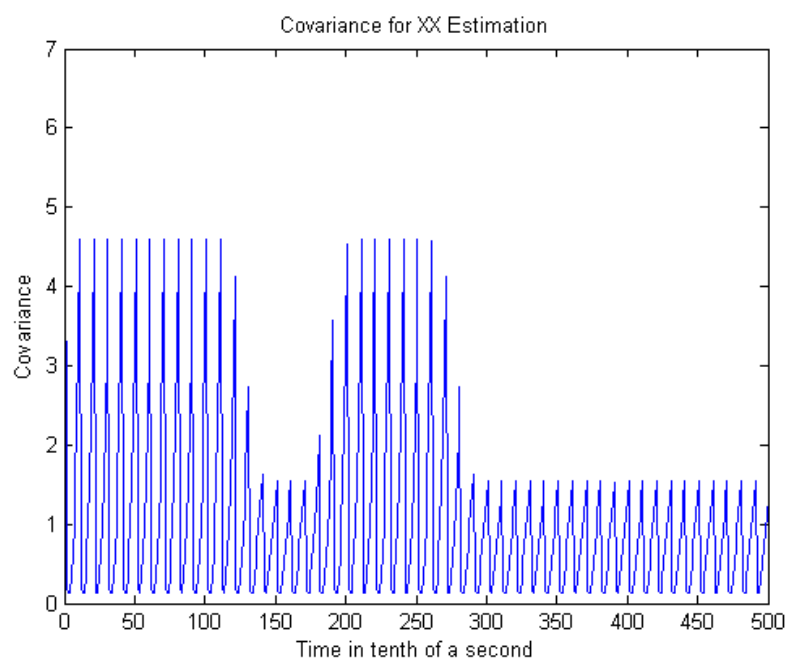


Figura 4.55: Covariâncias da Posição do Robot XX - Cenário 2

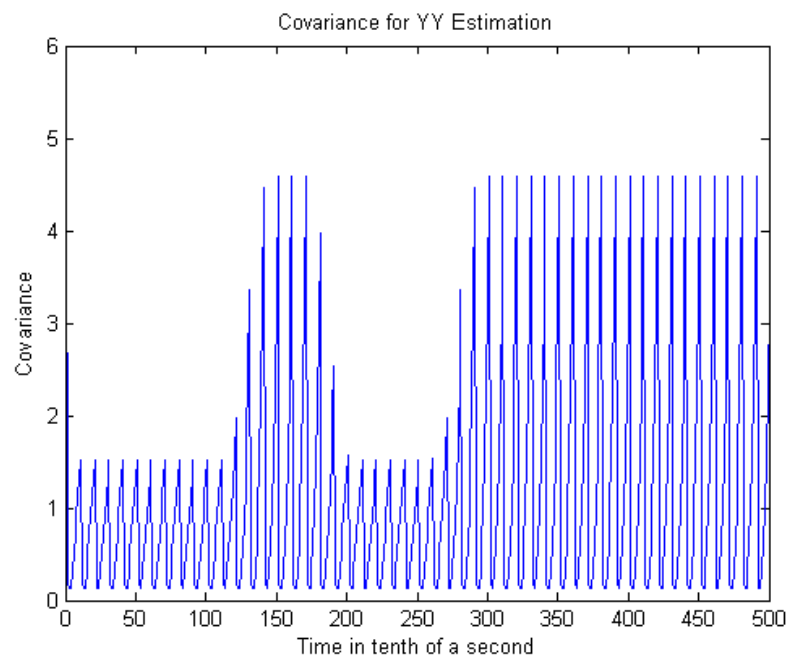


Figura 4.56: Covariâncias da Posição do Robot YY - Cenário 2

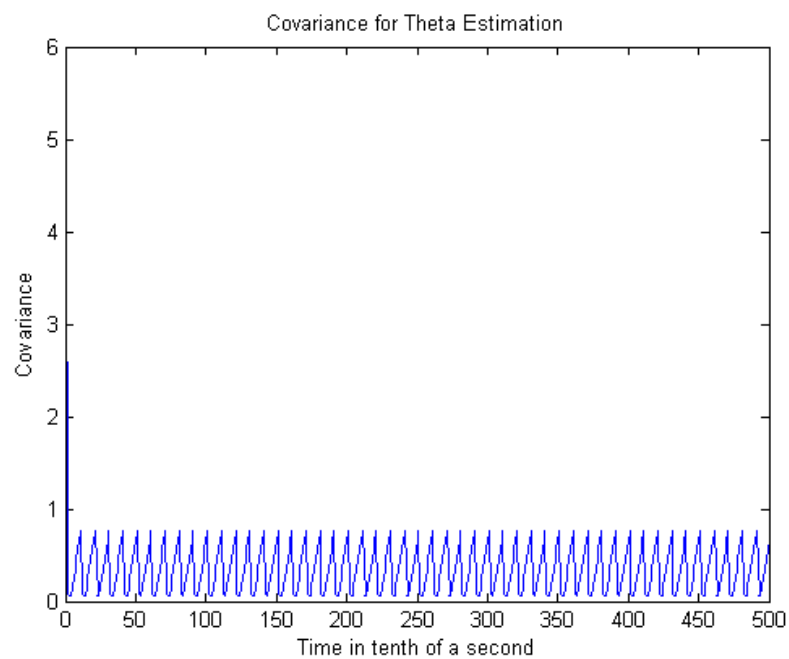


Figura 4.57: Covariâncias da Posição do Theta do Robot - Cenário 2

Na figura 4.54 podemos verificar que, no momento em que ocorrem os raptos, a estimativa e a posição real do robot não coincidem. O que é natural, já que em termos de modelo um “teleporte” puro não pode existir. Com a chegada de medidas de boa qualidade, a estimativa volta a recuperar convergindo de novo para a posição real.

Apresentamos agora algumas das imagens da movimentação do robot:

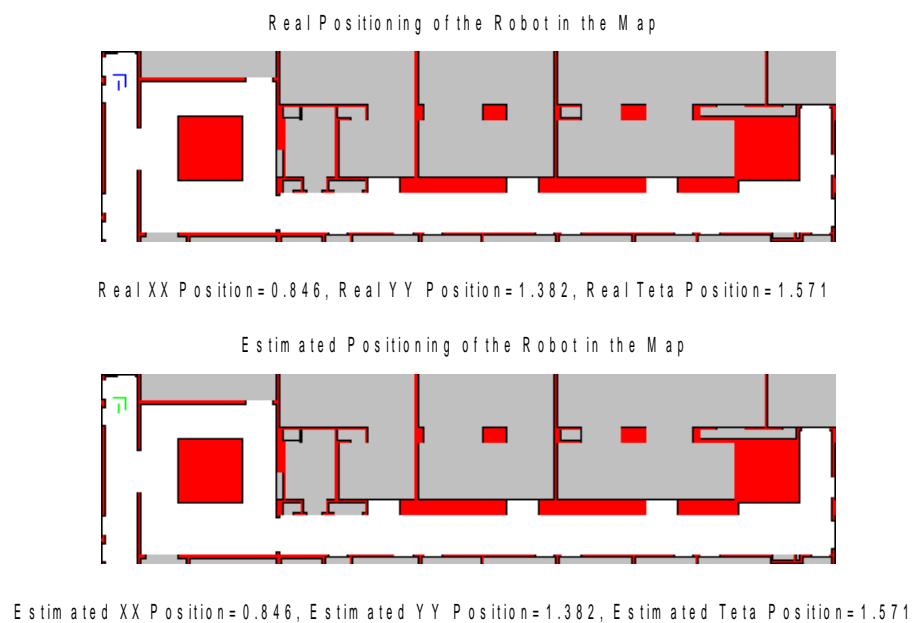


Figura 4.58: Representação simulada no mapa - Cenário 2

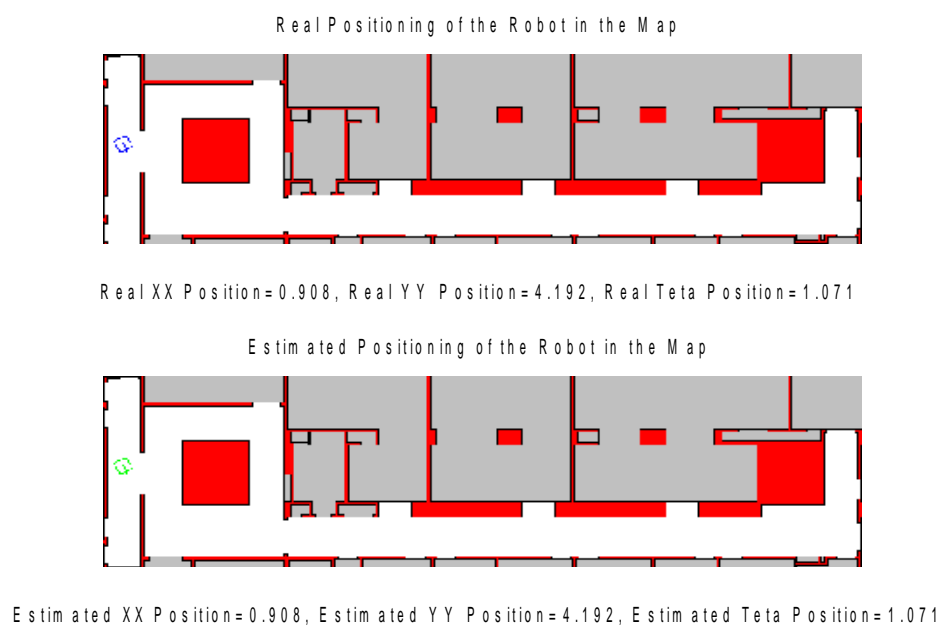


Figura 4.59: Representação simulada no mapa - cenário 2

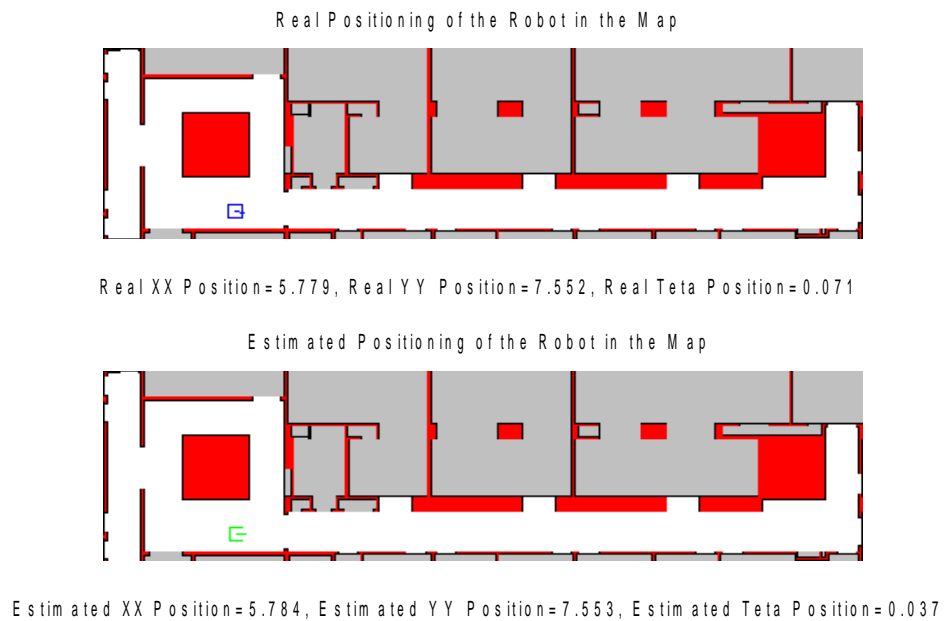


Figura 4.60: Representação simulada no mapa - cenário 2

Como podemos verificar, nem existem momentos nesta simulação em que a posição real do robot e a posição estimada pelo Filtro de Kalman não coincidam 4.60, mas esta situação é pontual e ocorre unicamente por breves instantes e logo após o rapto do robot. Não podendo retirar, desta situação, que o Filtro de Kalman não é um bom estimador, pois tal conclusão estaria completamente errada.

Das simulações retiradas, podemos concluir a qualidade da estimação realizada pelo Filtro de Kalman, da sua leveza computacional em comparação a outros métodos de estimação, também bastante utilizados para este fim.

4.6.3 Dados Experimentais

Fusão de Informação da Odometria

Considere-se a figura 4.61 onde se mostra o robot de tracção diferencial em movimento, onde assume que não existe escorregamento de qualquer tipo, nomeadamente lateral, ou seja, que a velocidade das rodas, no ponto de contacto com o chão é sempre perpendicular ao seu eixo.[14]

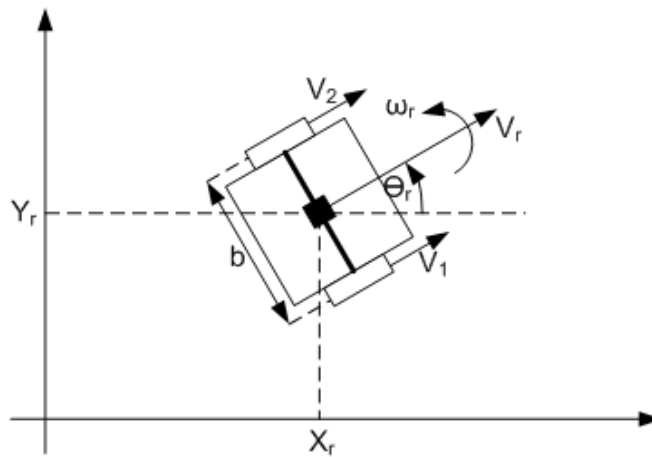


Figura 4.61: Representação do Robot em Movimentação pelo Mundo

O vector de estado escolhido para o sistema é:

$$X(t) = [x_r(t) \ y_r(t) \ \theta_r(t) \ v_r(t) \ \omega_r(t)]^T \quad (4.16)$$

onde (x_r, y_r) corresponde a posição cartesiana do robot e θ_r a sua direcção; v_r a velocidade tangencial e ω_r a velocidade angular. Como se assume que o robot é de tracção diferencial e sem escorregamento a componente normal da velocidade é sempre nula.

O movimento é então modelizado através da velocidade de cada roda, v_1 e v_2 e como tal verifica-se que:

$$\begin{cases} v_t = \frac{(v_1 + v_2)}{2} \\ \omega_r = \frac{(v_1 - v_2)}{b} \end{cases} \quad (4.17)$$

onde b é a distância entre rodas do robot.

O sistema de odometria fornece valores que são convertidos directamente em velocidades de cada roda v_1 e v_2 . Como os valores de v_r e ω_r são

facilmente obtidos e apresentam variações lentas pode considerar-se que estes são entradas de comando e assim a equação de cinemática do robot no tempo contínuo é:

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} v(t) \cdot \cos(\theta(t)) \\ v(t) \cdot \sin(\theta(t)) \\ \omega(t) \end{bmatrix} \quad (4.18)$$

No sentido de obter a equação de actualização do sistema em tempo discreto considera-se que os sinais de controlo são constantes entre cada instante de amostragem. De acordo com as equações 3.70 pág.30 e 3.87 pág.33 a dinâmica em tempo discreto do sistema é:

$$\begin{aligned} X(t_k) &= A X(t_{k-1}) + B u(t_{k-1}) \\ y(t_k) &= H x(t_k) \end{aligned} \quad (4.19)$$

E a matriz de estado na sua versão discreta fica:

$$A_k = \begin{bmatrix} 0 & 0 & -v(t_k) \sin(\theta(t_k)) \\ 0 & 0 & v(t_k) \cos(\theta(t_k)) \\ 0 & 0 & 0 \end{bmatrix} \quad (4.20)$$

$$A = \exp(A(k)(t(k) - t(k-1)))$$

Em termos do modelo, para o erro correspondente ao modelo dinâmico do sistema, este é assumido como, o modelo do erro da odometria, já que o modelo é baseado nesta característica.

$$Q = \begin{bmatrix} \sigma_{x_r}^2 & 0 & 0 \\ 0 & \sigma_{y_r}^2 & 0 \\ 0 & 0 & \sigma_{\theta_r}^2 \end{bmatrix} \quad (4.21)$$

[15]

Nas figuras 4.65, 4.62, 4.63 é possível verificar os dados reais obtidos quando se considera somente a odometria, ou seja, quando unicamente, é realizada a fase de previsão do Filtro de Kalman.

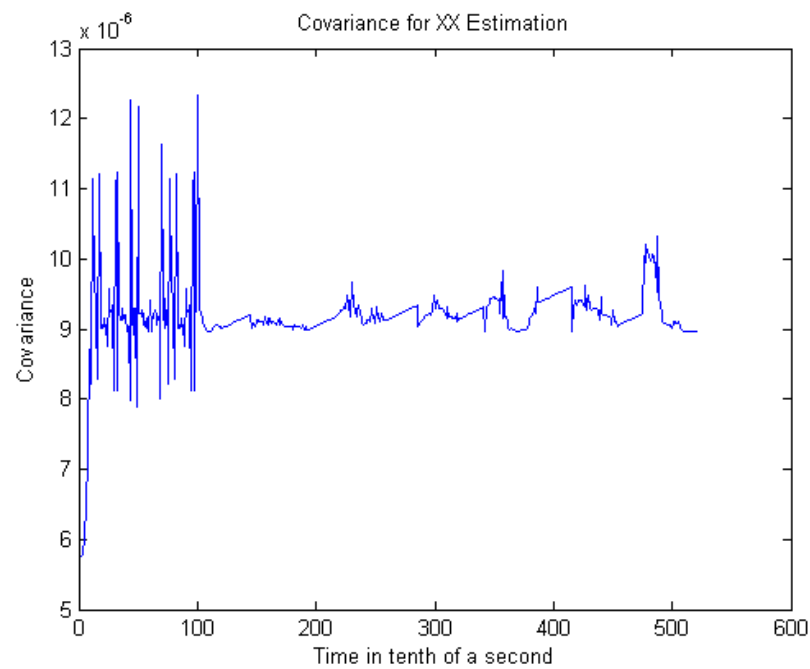


Figura 4.62: Covariância da Estimativa da Posição do Robot em XX - Dados Práticos de Odometria

Em termos de confiança em XX podemos verificar uma maior oscilação dos valores iniciais, o que pode ser explicado pelo tipo de movimentação, já que nesses instantes existe muita oscilação sobre o ponto em XX. Quando a movimentação em XX se torna mais constante a covariância estabiliza um pouco.

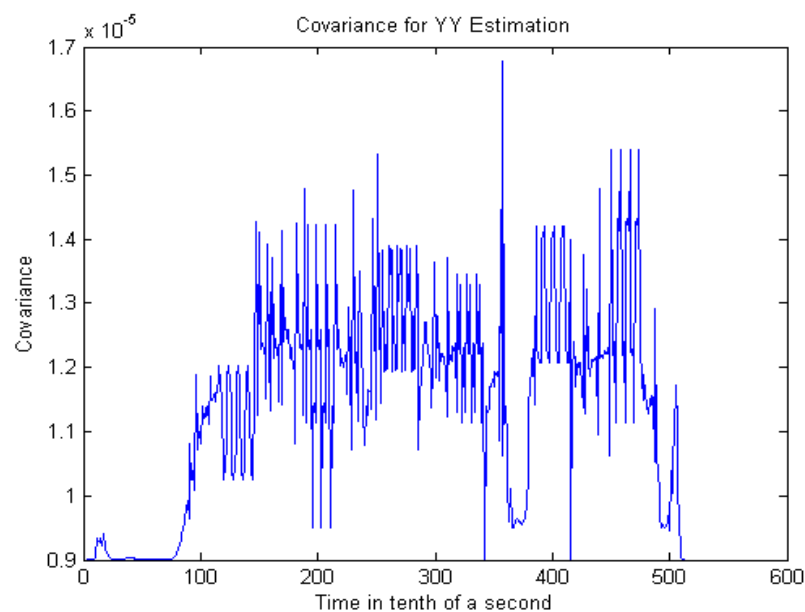


Figura 4.63: Covariância da Estimativa da Posição do Robot em YY - Run de Odometria

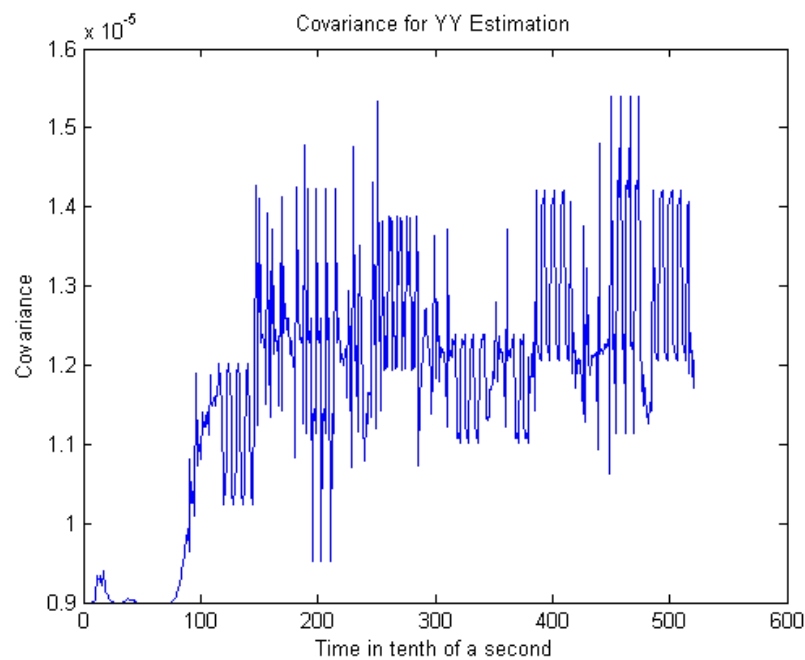


Figura 4.64: Covariância da Estimativa da Posição do Robot em YY - Dados Práticos de Odometria

Na figura 4.64 é possível confirmar as conclusões tiradas da figura 4.62, já que aqui, a partir dos 10seg o Robot de Limpeza começa a andar segundo o eixo dos XX, o que torna o valor do YY oscilante em termos de um mesmo valor, o que provoca oscilações nos valores da covariância.

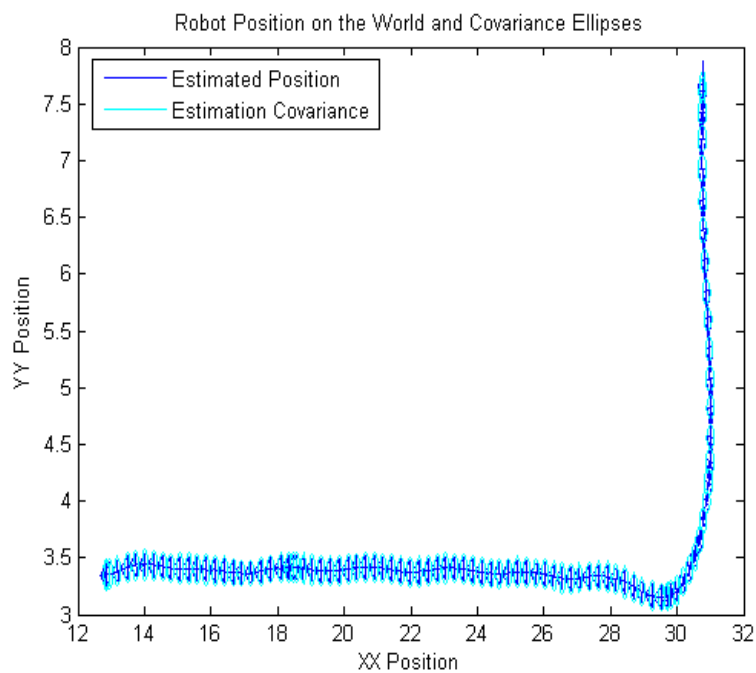


Figura 4.65: Representação da Estimativa da Posição do Robot e Covariâncias Associadas - Dados Práticos de Odometria

Por fim, temos imagens do "run" em termos de posição absoluta, figura 4.65. Se considerarmos somente a imagem, podíamos dizer que o robot de limpeza se portou bem, ou seja, que realizou as trajectórias desejadas, mas na realidade não é verdade, pois em termos de posição real, o robot não se encontra na posição definida, isto pode ser explicado devido a característica acumulativa do erro da odometria.

Em suma, utilizar unicamente odometria para estimar a posição real do robot não é suficiente, pois mesmo achando que se encontra no sentido correcto e a realizar uma trajectória perfeita, esta situação nunca é conseguida.

Fusão de Informação do Leitor de Códigos de Barras

Uma das informações a fundir, corresponde às distâncias medidas aos códigos de barras, que se encontram em posições definidas na zona de acção do robot. Considerando a figura 4.66 como a representação esquemática do robot ao avistar um código de barras:

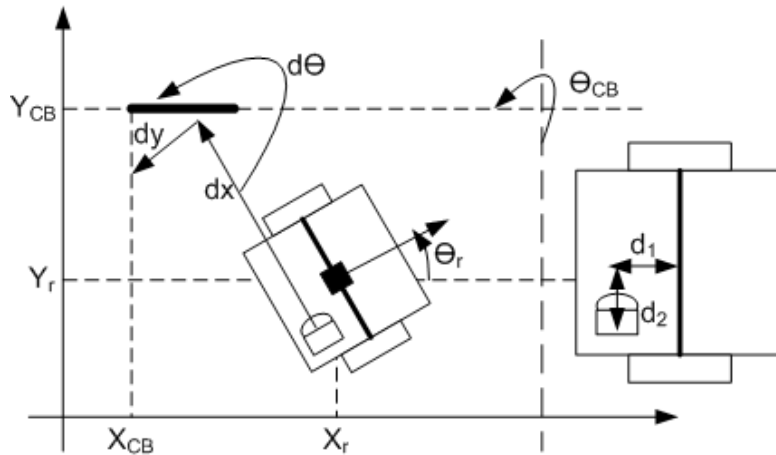


Figura 4.66: Representação do Robot a avistar um Código de Barras

É possível determinar a posição do robot, baseado nas medidas enviadas pelo programa de análise e localização de código de barras como:

$$Pos_{BC} = \begin{bmatrix} X_{BC} & Y_{BC} & \theta_{BC} \end{bmatrix}^T$$

$$Pos_R = \begin{bmatrix} X_r & Y_r & \theta_r \end{bmatrix}^T \quad (4.22)$$

$$Pos_{BC-R} = \begin{bmatrix} dx & dy & d\theta \end{bmatrix}^T$$

$$\begin{cases} X_r = X_{BC} + dx \cos(\theta_{BC}) + dy \sin(\theta_{BC}) - d_1 \sin(\theta_{BC}) - d_2 \cos(\theta_{BC}) \\ Y_r = Y_{BC} + dx \sin(\theta_{BC}) + dy \cos(\theta_{BC}) + d_1 \cos(\theta_{BC}) - d_2 \sin(\theta_{BC}) \\ \theta_r = \theta_{CB} + \pi - d\theta \end{cases} \quad (4.23)$$

Se considerarmos, não as medidas enviadas pelo programa de localização de códigos de barras, mas sim as projecções dessas medidas no referencial do mundo e para o centro do robot, ficamos com as seguintes equações:

$$\begin{cases} X_r = X_{BC} + dx' \\ Y_r = Y_{BC} + dy' \\ \theta_r = \theta_{CB} + d\theta' \end{cases} \Leftrightarrow \begin{cases} dx' = X_r - X_{BC} \\ dy' = Y_r - Y_{BC} \\ d\theta' = \theta_r - \theta_{CB} \end{cases} \quad (4.24)$$

onde:

$$\begin{cases} dx' = dx \cos(\theta_{BC}) + dy \sin(\theta_{BC}) - d_1 \sin(\theta_{BC}) - d_2 \cos(\theta_{BC}) \\ dy' = dx \sin(\theta_{BC}) + dy \cos(\theta_{BC}) + d_1 \cos(\theta_{BC}) - d_2 \sin(\theta_{BC}) \\ d\theta' = \pi - d\theta \end{cases} \quad (4.25)$$

Tendo em consideração as equações das medidas apresentadas em 4.24 é necessário agora calcular o jacobiano da medida:

$$H^{BC} = \frac{\partial(dx', dy', d\theta')}{\partial(X_r, Y_r, \theta_r)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.26)$$

Em termos do modelo do erro da medida, este é dado pelas variâncias das medidas:

$$R^{BC} = \begin{bmatrix} \sigma_{x_{BC}}^2 & 0 & 0 \\ 0 & \sigma_{y_{BC}}^2 & 0 \\ 0 & 0 & \sigma_{\theta_{BC}}^2 \end{bmatrix} \quad (4.27)$$

Considera-se que os erros da medida, obtidos pela câmara no seu referencial, são os mesmos que quando referenciados ao centro do robot no referencial do mundo, havendo somente a necessidade de realizar a rotação entre os referenciais.

Sendo assim:

$$V \cdot R \cdot V^T = \begin{bmatrix} \sigma_{dx_{BC}}^2 \cos(\theta_{CB}) & \sigma_{dy_{BC}}^2 \sin(\theta_{CB}) & 0 \\ \sigma_{dx_{BC}}^2 \sin(\theta_{CB}) & \sigma_{dy_{BC}}^2 \cos(\theta_{CB}) & 0 \\ 0 & 0 & \sigma_{d\theta_{BC}}^2 \end{bmatrix} \quad (4.28)$$

Devido ao facto de a localização por código de barras ser considerado eminente no local, não existem problemas relativos à localização de funções periódicas na relação 4.25. Na implementação do EKF no robot, inclui-se um teste de verosimilhança para as medidas, utilizando para esse facto as informações adicionais obtidas pelo programa de localização por códigos de barras. Sendo assim, só são consideradas medidas válidas, se o número de linhas processadas for superior a duas.

Nas figuras 4.67, 4.68, 4.69 é possível verificar os dados reais obtidos, quando se considera os dados da odometria na realização da previsão e os dados sobre a localização por códigos de barras para fase de actualização.

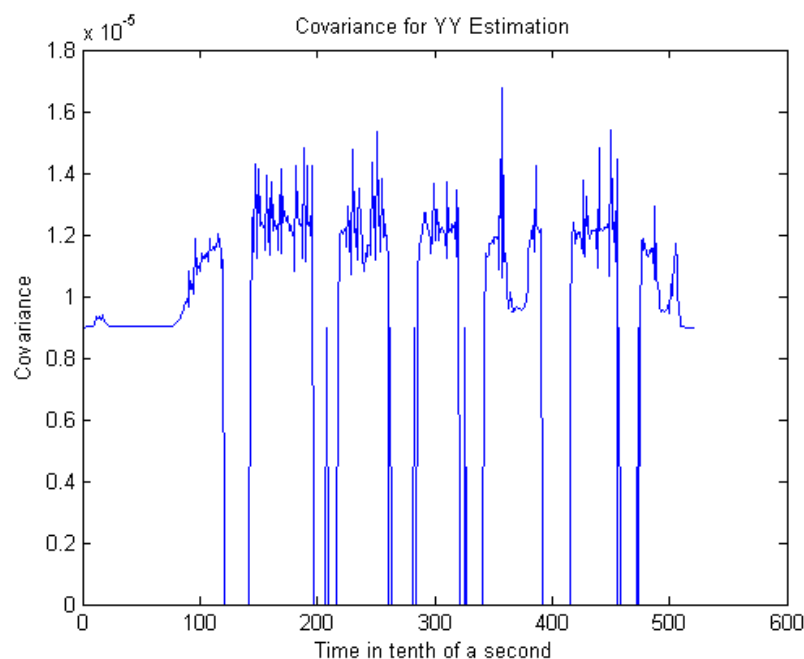


Figura 4.68: Covariância da Estimativa da Posição do Robot em YY - Dados Práticos com Localização por Códigos de Barras

Como no caso anterior, onde só, se realizava a fase de previsão, o Filtro de Kalman Extendido, sempre que o robot se movimentava sobre uma das coordenadas cartesianas, a outra, apresentava em termos de covariância, um valor mais oscilatório.

Nas figuras 4.67 e 4.68 podemos aperceber-nos dos momentos em que a fase de actualização é realizada, pois os valores da covariância tanto em X como em Y são bastante diminuídos. Havendo uma convergência quase imediata para a posição estimada do Robot de Limpeza, baseada nos dados do programa de localização por códigos de barras.

Os valores para a covariância são muito diminuídos, sempre que existe informação da localização por códigos de barras, pois, devido à qualidade da medida e à sua pequena frequência, foi colocado um ganho elevado, pois assumiu-se que o erro da camera em relação ao erro da odometria era muito mais pequeno, o que provocou que quando o Robot de Limpeza, avista-se um código de barras o valor do ganho de Kalman se aproximar bastante de H^{-1} .

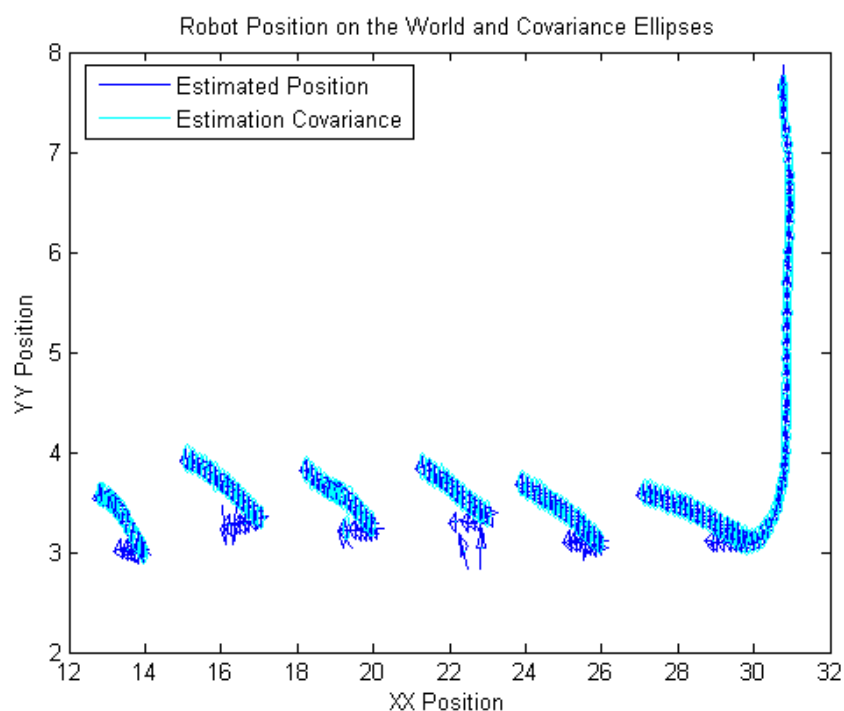


Figura 4.69: Representação da Estimativa da Posição do Robot e Covariâncias Associadas - Dados Práticos com Localização por Códigos de Barras

Da figura 4.69 podemos verificar que não apresenta a mesma disposição linear de trajectória, como a representada na figura 4.65, com esta situação podemos concluir, que a característica acumulativa do erro da odometria é trágico, necessitando de constante acerto de outro tipo de informação de modo a diminuir o efeito acumulativo desse erro.

Em termos reais, a posição estimada pelo Robot de Limpeza e a sua posição real no ambiente onde se encontra inserido, foi agora muito mais próxima, demonstrando de forma inequívoca às características de fusão de informação do Filtro de Kalman, acima de tudo a sua necessidade quase absoluta quando se pretende obter uma estimativa adequada para a posição em sistemas robóticos.

Fusão de Informação das Sonares

Outra fonte de localização corresponde à informação obtida a partir da medição realizada pelos sonares. Na figura 4.70 encontra-se representado o esquema das medidas realizadas.

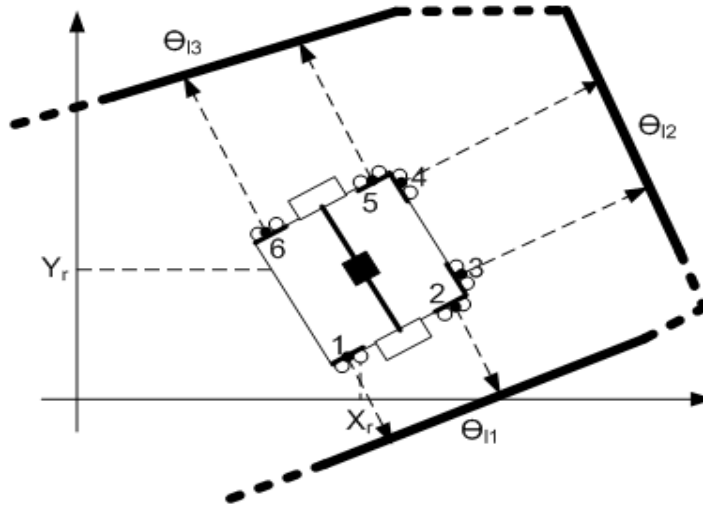


Figura 4.70: Representação do Robot a realizar medidas com os Sonares

Com as medidas dos sonares, obtemos seis distâncias e é possível determinar três ângulos.

Os ângulos são calculados do seguinte modo:

$$\theta_{1,2} = \arctan\left(\frac{d_1 - d_2}{\text{dist}_{1,2}}\right); \quad \theta_{3,4} = \arctan\left(\frac{d_3 - d_4}{\text{dist}_{3,4}}\right); \quad \theta_{5,6} = \arctan\left(\frac{d_5 - d_6}{\text{dist}_{5,6}}\right). \quad (4.29)$$

Por forma a não fundir duas vezes a mesma informação, somente utilizamos uma das distâncias medidas de cada lado, é utilizada assim como o ângulo determinado, escolhendo-se sempre a distância mínima entre os dois. Sendo assim, os vectores das medidas a fundir são:

$$Y_r^s = \begin{bmatrix} \min(d_1; d_2) \\ \alpha_{1,2} \end{bmatrix}; \quad Y_f^s = \begin{bmatrix} \min(d_3; d_4) \\ \alpha_{3,4} \end{bmatrix}; \quad Y_l^s = \begin{bmatrix} \min(d_5; d_6) \\ \alpha_{5,6} \end{bmatrix}. \quad (4.30)$$

Tendo em consideração o apresentado em 4.30, podemos determinar o modelo das medidas como:

$$\begin{cases} \theta_w = \theta_s + \theta_r \\ d_w = d_s - x_r \cdot \sin(\theta_w) + y_r \cdot \cos(\theta_w) \end{cases} \quad (4.31)$$

Colocando em ordem as medidas dos sonares:

$$\begin{cases} \theta_s = \theta_w - \theta_r \\ d_s = d_w + x_r \cdot \sin(\theta_w) - y_r \cdot \cos(\theta_w) \end{cases} \quad (4.32)$$

Determinando o jacobiano da medida:

$$H^{Sonar} = \frac{\partial(d_s, \theta_s)}{\partial(x_r, y_r, \theta_r)} = \begin{bmatrix} 0 & 0 & -1 \\ \sin(\theta_w) & \cos(\theta_w) & 0 \end{bmatrix} \quad (4.33)$$

Em termos do erro as medidas, são dadas pela variância dos erros dos sonares:

$$R^{Sonar} = \begin{bmatrix} \sigma_{d_s}^2 & 0 \\ 0 & \sigma_{\theta_s}^2 \end{bmatrix} \quad (4.34)$$

Considera-se que em termos do erro, não existe correlação entre a medida da distância e a medida do ângulo.

Devido ao facto de a determinação da distância e do ângulo corresponderem a medidas eminentemente locais, não existem problemas relativos à utilização de funções periódicas na relação 4.32.

Na implementação do Filtro de Kalman Extendido no Robot de Limpeza, um filtro de verosimilhança foi aplicado para cada um dos conjuntos de medida, onde se testa o valor de cada uma das medidas, afim de se validar ou não, o ângulo calculado. No caso dos sonares da frente mais uma acção de teste é realizada, onde se compara o ângulo determinado pelos sonares da frente, com os ângulos determinados de cada uma das outras faces.

Em termos do estado da implementação prática, tudo o que é afirmado encontra-se implementado no Robot de Limpeza mas, infelizmente até a altura da escrita deste capítulos não foi possível apresentar os resultados práticos, já que ainda se encontra em fase de teste.

Fusão de Informação dos Sharps

Por fim, temos a última fonte de localização, que corresponde a informação obtida a partir da medição realizada pelo sharp, na figura 4.71 encontra-se representada o esquema das medidas realizadas.

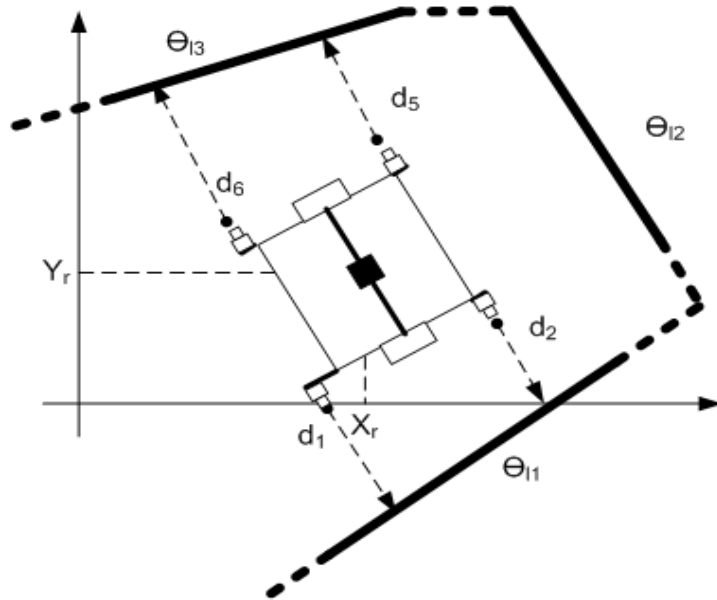


Figura 4.71: Representação do Robot a obter medidas dos Sharps

Em relação ao tipo de medidas, este é muito semelhante ao obtido com os sonares, mas com muito menor erro e interferência. Uma outra diferença, corresponde ao número de sharps presentes, já que não se encontram montados neste momento sharps na face frontal do Robot de Limpeza.

Os ângulos são calculados do seguinte modo:

$$\theta_{1,2} = \arctan\left(\frac{d_1 - d_2}{dist_{1,2}}\right); \quad \theta_{5,6} = \arctan\left(\frac{d_5 - d_6}{dist_{5,6}}\right). \quad (4.35)$$

Por forma a não fundir duas vezes a mesma informação, somente uma das distâncias medidas para cada um dos lados, assim como o ângulos determinado, escolhendo-se sempre à distância mínima entre obtidas dois. Sendo assim, os vectores das medidas a fundir são:

$$Y_r^s = \begin{bmatrix} \min(d_1; d_2) \\ \alpha_{1,2} \end{bmatrix}; \quad Y_l^s = \begin{bmatrix} \min(d_5; d_6) \\ \alpha_{5,6} \end{bmatrix}. \quad (4.36)$$

Tendo em consideração o apresentado em 4.35, podemos determinar o modelo das medidas como:

$$\begin{cases} \theta_w = \theta_{sharp} + \theta_r \\ d_w = d_{sharp} - x_r \cdot \sin(\theta_w) + y_r \cdot \cos(\theta_w) \end{cases} \quad (4.37)$$

Colocando em ordem as medidas dos sonares:

$$\begin{cases} \theta_{sharp} = \theta_w - \theta_r \\ d_{sharp} = d_w + x_r \cdot \sin(\theta_w) - y_r \cdot \cos(\theta_w) \end{cases} \quad (4.38)$$

Determinando o jacobiano da medida:

$$H^{Sharp} = \frac{\partial(d_s, \theta_s)}{\partial(x_r, y_r, \theta_r)} = \begin{bmatrix} 0 & 0 & -1 \\ \sin(\theta_w) & \cos(\theta_w) & 0 \end{bmatrix} \quad (4.39)$$

Em termos do erro das medidas este é dado pela variância dos erros dos sharps. Estes são:

$$R^{Sharp} = \begin{bmatrix} \sigma_{d_s}^2 & 0 \\ 0 & \sigma_{\theta_s}^2 \end{bmatrix} \quad (4.40)$$

Considera-se que em termos do erro, não existe correlação entre a medida da distância e a medida do ângulo.

Devido ao facto da determinação da distância e do ângulo corresponderem a medidas eminentemente locais, não existem problemas relativos à utilização de funções periódicas na relação 4.36.

Na implementação do Filtro de Kalman Extendido no Robot de Limpeza, um filtro de verosimilhança foi aplicado para cada um dos conjuntos de medida, onde se testa o valor de cada uma das medidas, afim de se validar ou não o ângulo calculado. Essa acção nos sharps é importante, devido à gama de medidas linear restrita que estes apresentam.

Em termos do estado da implementação prática, tudo o que é afirmado encontra-se implementado no Robot de Limpeza mas, infelizmente até a altura da escrita deste capítulos não foi possível apresentar os resultados práticos, já que ainda se encontra em fase de teste.

4.7 Conclusões

Este capítulo reúne todo o trabalho experimental realizado no Robot de Limpeza.

Após a apresentação do Robot de Limpeza como um Sistema Robótico autónomo, apresentou-se as diversas áreas em que o Robot de Limpeza foi trabalhado e estudado.

Na fase exactamente seguinte, foram apresentadas as diversas calibração realizadas, assim como, os modelos do erro para cada uma delas.

Primeiro com a calibração dos PID's, numa tentativa de estabilizar a resposta dos motores, em relação a uma referência em degrau. De seguida, realizou-se a calibração da odometria, onde se procurou definir de uma forma experimental o valor da distância entre rodas e do perímetro das mesmas. O conhecimento destas constantes são fulcrais na conversão das medidas obtidas pelos encoders para as medidas SI correspondentes. Após estas calibrações foi também determinado o modelo do erro da odometria, que será utilizado na implementação do Filtro de Kalman.

De seguida realizou-se a apresentação da arquitectura de software utilizada. Arquitectura esta, que se comprovou ser de fácil entendimento e aberta, ou seja, que pode ser alterada, de modo a acrescentar novas funcionalidades ou para aquisição de novos sensores, com potencialidade de criação de um histórico, que poderá ser analisado ao pormenor e "offline" numa aplicação especialmente desenvolvida para esse efeito.

Por fim, foi analisada a problemática da auto-localização robótica com a aplicação do Filtro de Kalman. Iniciando um estudo preliminar baseado em simulações em Matlab passando depois para o desenvolvimento real na plataforma do Robot de Limpeza.

Em termos da fusão de informação na plataforma do Robot de Limpeza, foi implementado o Filtro de Kalman Extendido, que utiliza os dados da odometria para realizar a previsão e as medidas sensoriais provenientes da localização por código de barras, sonares e sharps na fase de actualização.

Infelizmente até a data da entrega da versão provisória da dissertação, somente a actualização baseada em localização de código de barras tinha sido devidamente testada, encontrando-se a fusão com a informação dos sharps e dos sonares ainda em testes.

Mesmo assim, foi possível concluir que só realizando fusão de informação é que é possível apresentar uma estimativa coerente e correcta para a posição global do Robot de Limpeza. Com um erro acumulativa para a odometria a posição estimada e real divergiam, caso em que só foi realizada a fase de previsão do Filtro de Kalman, somando-lhe a informação obtida pela localização

por códigos de barras, a estimativa era actualizada devidamente, diminuindo a influência do erro da odometria, o que promoveu um "run" em que no final a posição estimada e a posição real fossem na realidade muito próximas.

Tendo isto em mente, e conhecendo o tipo de medidas obtidas tanto pelos sonares como pelos sharps, no caso da realização da fusão de informação com todos estes dados, a qualidade da estimação iria melhorar imenso. Especialmente, porque esta informação proveniente dos sonares e sharps é muito mais frequente do que os dados de localização por código de barras, esperando-se obter duas distâncias e dois ângulos por cada ciclo de controlo, tanto para os sonares como para os sharps.

Será no caso da fusão de toda esta informação, que o potencial de estimação do Filtro de Kalman Extendido é capaz de demonstrar todo o seu valor, fundindo informações diferentes, unificando-as numa estimativa única de melhor qualidade que a soma das estimativa realizadas por cada um dos tipos de informação sensorial obtidos quando analisados sozinhos.

Capítulo 5

Conclusões

5.1 Conclusões Gerais

Esta dissertação aborda a problemática da autonomia e auto-localização num sistema robótico móvel de serviço.

Sendo o tema da auto-localização o centro desta dissertação, foi discutida esta problemática e apresentado de um modo formal, o Filtro de Kalman e algumas das suas extensões mais importantes, mas também, a sua implementação prática no Robot de Limpeza.

Os aspectos mais relevantes relativos à aplicação prática desenvolvida são:

- A definição e implementação de um sistema de controlo robusto com possibilidade de ampliação do sistema com a definição e implementação de novos algoritmos de auto-localização ou introdução de mais dados sensoriais tanto por via do aumento do número dos presentes ou mesmo a introdução de novos tipos de sensores;
- Projecto, simulação e implementação de um Filtro de Kalman Extendido (EKF) para fusão sensorial dos dados disponíveis. A convergência deste

tipo de filtros não é assegurada. As limitações inerentes a cada tipo de medida foi referida e foram confirmados os benefícios da utilização de vários tipos de medida para a convergência do filtro. Tendo sido verificado experimentalmente, a convergência do método quando ocorre a fusão com as medidas disponíveis. A correcção de cada medida depende, no entanto, de testes de verosimilhança de modo a que não seja fundida, informação não válida;

- De modo a facilitar a localização recomenda-se:
 - a utilização de códigos de barras para a determinação da posição global do robot, mas estes apresentam problemas de visibilidade, estando dependentes da distância da câmara ao código de barras e ainda da luminosidade da área onde se encontram;
 - fusão de informação de diversos tipos de sensores, já que a soma de todas as contribuições é superior à contribuição das mesmas individualmente para a localização.
- Foi demonstrada a viabilidade prática do sistema de localização;
- Foi demonstrada a necessidade e facilidade proporcionada pela definição da arquitectura de controlo e a facilidade de utilização e análise de dados proporcionada pelas diferentes aplicações desenvolvidas.

De forma resumida, pode dizer-se que este trabalho explora várias áreas da robótica, apresentado as mais importantes questões de projecto. É estabelecida também a importância da auto-localização em sistemas robóticos autónomos e da facilidade de análise de dados quando o sistema robótico apresenta uma arquitectura de controlo bem estruturada e pensada.

5.2 Trabalho Futuro

Tomando que a implementação baseada no Filtro de Kalman Extendido encontra-se neste momento estável, desenvolvimento futuros deverão incidir na camadas de software superiores:

- introdução de mais sensores, de modo a expandir a capacidade de fusão de informação e de modo a tornar ainda mais robusta, a implementação desenvolvida;
- desenvolvimento das funções de visão, com reconhecimentos de características específicas do ambiente em que o robot se insere, por exemplo, detecção de portas ou dos aquecedores do corredor;

- desenvolver e estudar melhor uma implementação baseada em Filtro de Kalman Extendido Iterativo, especialmente em sensores com dados frequentes de modelo de erro com elevada covariância;
- melhoramento das funções de movimentação;
- definição e estudo de algoritmos de path planning;

Com este desenvolvimento o projecto de Robot de Limpeza irá se tornar ainda mais abrangente, passando a tocar em todas as áreas dos sistemas robóticos. Passando a ser uma ferramenta de inovação e desenvolvimento ainda mais reconhecida do que é hoje.

Referências Bibliográficas

- [1] André Gustavo Scolari Conceição, "Controlo e Cooperação de Robôs Móveis Autónomos Omnidireccionais", (2007)
- [2] Armando Jorge Miranda de Sousa, "Arquitecturas de Sisteamas Robóticos e Localização em Tempo Real Através de Visão. Aplicação no Domínio do Futebol Robótico", (2003)
- [3] C. E. Garcia and M. Morari, "Internal Model Control. 1. A Unifying Review and Some New Results", (1982)
- [4] I. L. Chien and P.S. Fruehauf, "Consider IMC tuning to improve controller performance, Chemical Engineering Progress", (1990)
- [5] R. E. Kalman, "A New Approach for to Linear Filtering and Prediction Problems", Transactions ASME Journal of Basic Engineering 35-45 (1960)
- [6] Norbert Wiener, "Extrapolation, Interpolation and Smoothing of Stationary Time Series", (1949)
- [7] M. Grewal, S. Mohinder and Angus P. Andrews, "Kalman Filtering The Theory and Practice", (1993)
- [8] Peter S. Maybeck, "Stochastic Models, Estimation and Control", (1979)

- [9] Kai O. Arras, José A. Castellanos, Martion Schilt, Roland Siegwart, "Feature-based multi-hypothesis localization and tracking using geometric constraints", (2003)
- [10] Rudy Negenborn, "Robot Localization and Kalman Filters. On finding your position in a noisy world", (2003)
- [11] Greg Welch, Gary Bishop, "An Introduction to the Kalman Filter", (2006)
- [12] Erik Cuevas, Daniel Zaldivar and Raul Rojas, "Kalman Filter for Vision Tracking", (2005)
- [13] Gustavo Gama da Rocha Pimentel, ""The Cleaning RobotProject" - Aplicação do Filtro de Partículas como sistema defusão de informação", (2008)
- [14] Paulo José Cerqueira Gomes da Costa, "Identificação, Modelização e Controlo de um Veículo Móvel Autónomo", (1995)
- [15] Armando J. M. Sousa, Paulo J. C. G. Costa and António P. G. M. Moreira, "Sistema de Localização de Robôs Móveis Baseado em Filtro de Kalman Extendido", (2001)

Anexo

Dados Experimentais da Calibração da Odometria

Tabela 5.1: Dados Experimentais para a formulação do Modelo da Odometria relativos à Translação e Deslizamento

Dados obtidos pelo Robot				Dados Reais			
ΔX	ΔY	ΔRho	$\Delta Theta$	ΔX	ΔY	ΔRho	$\Delta Theta$
1,1015	0,0322	1,1020	0,0292	1,1170	0,0350	1,1177	0,0358
1,0970	0,0206	1,0972	0,0188	1,1140	0,0200	1,1142	0,0180
1,1096	0,0217	1,1098	0,0196	1,1260	0,0200	1,1262	0,0178
1,0974	0,0268	1,0977	0,0244	1,1160	0,0290	1,1164	0,0260
1,0992	0,0229	1,0994	0,0208	1,1210	0,0200	1,1212	0,0178
1,0995	0,0227	1,0997	0,0206	1,1270	0,0250	1,1173	0,0224
1,0956	0,0226	1,0958	0,0206	1,1160	0,0170	1,1161	0,0152
1,1142	0,0228	1,1144	0,0205	1,1340	0,0220	1,1341	0,0106
1,0968	0,0224	1,0970	0,0204	1,1190	0,0170	1,1191	0,0152
1,0998	0,0219	1,1000	0,0199	1,1210	0,0210	1,1212	0,0187

Tabela 5.2: Erros na Translação

Erros de Translação		
Erro de X	Erro de Y	Erro de Rho
0,0155	0,0078	0,0157
0,0170	-0,0006	0,0170
0,0164	-0,0017	0,0164
0,0186	0,0022	0,0186
0,0218	-0,0029	0,0217
0,0175	0,0023	0,0175
0,0204	-0,0056	0,0203
0,0184	-0,0108	0,0196
0,0222	-0,0054	0,0221
0,0212	-0,0009	0,0212

Tabela 5.3: Erros de Deslizamento

Erros de Deslizamento
Erro de Theta
0,0066
-0,0008
-0,0018
0,0016
-0,0030
0,0017
-0,0054
-0,0099
-0,0052
-0,0012

Tabela 5.4: Dados Experimentais para a Formulação do Modelo da odometria Relativos à Rotação

<i>Deslocamento do Centro do Robot</i>		<i>Dados obtidos pelo Robot</i>	<i>Dados Reais</i>
ΔX	ΔY	$\Delta\theta$	$\Delta\theta$
-0,0081	0,0139	6,6350	6,6328
-0,0106	0,0121	6,6359	6,6347
-0,0080	0,0110	6,6395	6,6286
-0,0122	0,0127	6,6148	6,6117
-0,0138	0,0162	6,6753	6,6642
-0,0106	0,0157	6,6174	6,6065
-0,0085	0,0130	6,6433	6,6344
-0,0087	0,0130	6,6228	6,6135
-0,0083	0,0142	6,6433	6,6362
-0,0076	0,0103	6,6481	6,6406

Anexo

Ficha Técnica

*Escrito em Open Office 2.3.1 sob Windows
Fontes: Trebuchet MS, Verdana*

Muitas figuras desenhadas utilizando Microsoft Visio 2003

URL da página do Autor: <http://www.fe.up.pt/~ee01189>

Fernando Pedro Vieira Freitas Pinto
7 Março 2008